Red Hat Enterprise Virtualization 3.3 Technical Reference Guide

---

The Technical Architecture of Red Hat Enterprise Virtualization Environments

Zac Dover Steve Gordon Tim Hildred

Red Hat Enterprise Virtualization 3.3 Technical Reference Guide

The Technical Architecture of Red Hat Enterprise Virtualization Environments

Zac Dover
zdover@redhat.com

Steve Gordon
sgordon@redhat.com

Tim Hildred
thildred@redhat.com

Legal Notice

Copyright © 2014 Red Hat, Inc.

This document is licensed by Red Hat under the Creative Commons Attribution-ShareAlike 3.0 Unported License. If you distribute this document, or a modified version of it, you must provide attribution to Red Hat, Inc. and provide a link to the original. If the document is modified, all Red Hat trademarks must be removed.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, JBoss, MetaMatrix, Fedora, the Infinity Logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux ® is the registered trademark of Linus Torvalds in the United States and other countries.

Java ® is a registered trademark of Oracle and/or its affiliates.

XFS ® is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL ® is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js ® is an official trademark of Joyent. Red Hat Software Collections is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack ® Word Mark and OpenStack Logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

Abstract

This guide documents the concepts, components, and technologies used in a Red Hat Enterprise Virtualization environment.

Table of Contents

Preface **4**
1. Document Conventions 4
1.1. Typographic Conventions 4
1.2. Pull-quote Conventions 5
1.3. Notes and Warnings 6
2. Getting Help and Giving Feedback 6
2.1. Do You Need Help? 6
2.2. We Need Feedback! 7

Chapter 1. Introduction **8**
1.1. Red Hat Enterprise Virtualization Manager 8
1.2. Red Hat Virtualization Hypervisor 8
1.3. Features Requiring a Compatibility Upgrade to Red Hat Enterprise Virtualization 3.3 10

Chapter 2. Red Hat Enterprise Virtualization Architecture **12**
2.1. Interfaces for Accessing the Manager 12
2.2. Components that Support the Manager 13
2.3. Storage 14
2.4. Network 15

Chapter 3. Storage architecture **19**
3.1. Storage Concepts 19
3.1.1. Data Centers 19
3.1.2. Storage Domains Overview 19
3.1.3. Types of Storage Backing Storage Domains 19
3.1.4. Storage Domain Types 20
3.1.5. Storage Formats for Virtual Machine Disk Images 21
3.1.6. Virtual Machine Disk Image Storage Allocation Policies 21
3.1.7. Storage Domain Autorecovery in Red Hat Enterprise Virtualization 22
3.1.8. The Storage Pool Manager 22
3.1.9. Storage Pool Manager Selection Process 24
3.1.10. Exclusive Resources and Sanlock in Red Hat Enterprise Virtualization 24
3.2. Block based storage provisioning mechanics 25
3.2.1. Thin Provisioning and Storage Over-Commitment 25
3.2.2. Logical Volume Extension 26

Chapter 4. Network architecture **28**
4.1. Network Architecture 28
4.2. Basic networking 28
4.2.1. Introduction: Basic Networking Terms 28
4.2.2. Network Interface Controller (NIC) 28
4.2.3. Bridge 28
4.2.4. Bond 29
Bonding Modes 29
4.2.5. Switch Configuration for Bonding 30
4.2.6. Virtual Network Interface Controller (VNIC) 30
4.2.7. Virtual LAN (VLAN) 32
4.3. Networking in data centers and clusters 32
4.3.1. Cluster Networking 32
4.3.2. Logical Networks 33
4.3.3. Required Networks, Optional Networks, and Virtual Machine Networks 35
4.4. Networking in hosts and virtual machines 36
4.4.1. Virtual Machine Connectivity 36

4.4.1. Virtual Machine Connectivity 36
4.4.2. Port Mirroring 36
4.4.3. Host Networking Configurations 36
4.4.4. Bridge Configuration 37
4.4.5. VLAN Configuration 37
4.4.6. Bridge and Bond Configuration 38
4.4.7. Multiple Bridge, Multiple VLAN, and NIC Configuration 39
4.4.8. Multiple Bridge, Multiple VLAN, and Bond Configuration 39

Chapter 5. Power management and fencing 41
5.1. Introduction to Power Management and Fencing 41
5.2. Power Management by Proxy in Red Hat Enterprise Virtualization 41
5.3. Power Management 41
5.4. Fencing 42
5.5. Soft-Fencing Hosts 43
5.6. Using Multiple Power Management Fencing Agents 44

Chapter 6. Load Balancing, scheduling, and migration 45
6.1. Load Balancing, Scheduling, and Migration 45
6.2. Load Balancing Policy 45
6.3. Load Balancing Policy: None 45
6.4. Load Balancing Policy: Even Distribution 46
6.5. Load Balancing Policy: Power Saving 46
6.6. Scheduling 46
6.7. Migration 47

Chapter 7. Directory services 48
7.1. Directory Services 48
7.2. Local Authentication: Internal Domain 48
7.3. Remote Authentication Using GSSAPI 49

Chapter 8. Templates and pools 50
8.1. Templates and Pools 50
8.2. Templates 50
8.3. Pools 51

Chapter 9. Virtual Machine Snapshots 52
9.1. Live Snapshots in Red Hat Enterprise Virtualization 52
9.2. Snapshot Creation 53
9.3. Snapshot Previews 55
9.4. Snapshot Deletion 55

Chapter 10. Hardware drivers and devices 57
10.1. Virtualized Hardware 57
10.2. Stable Device Addresses in Red Hat Enterprise Virtualization 57
10.3. Central Processing Unit (CPU) 58
10.4. System Devices 58
10.5. Network Devices 58
10.6. Graphics Devices 59
10.7. Storage Devices 59
10.8. Sound Devices 59
10.9. Serial Driver 59
10.10. Balloon Driver 60

Chapter 11. Technical minimums and limitations 61
11.1. Minimum Requirements and Supported Limits 61

11.2. Data Center Limitations 61
11.3. Cluster Limitations 61
11.4. Storage Domain Limitations 61
11.5. Red Hat Enterprise Virtualization Manager Limitations 63
11.6. Hypervisor Requirements 64
11.7. Guest Requirements and Support Limits 68
11.8. SPICE Limitations 69

Chapter 12. Additional References 70
12.1. Additional References 70

Revision History 71

Preface

1. Document Conventions

This manual uses several conventions to highlight certain words and phrases and draw attention to specific pieces of information.

1.1. Typographic Conventions

Four typographic conventions are used to call attention to specific words and phrases. These conventions, and the circumstances they apply to, are as follows.

`Mono-spaced Bold`

Used to highlight system input, including shell commands, file names and paths. Also used to highlight keys and key combinations. For example:

> To see the contents of the file **`my_next_bestselling_novel`** in your current working directory, enter the **`cat my_next_bestselling_novel`** command at the shell prompt and press **`Enter`** to execute the command.

The above includes a file name, a shell command and a key, all presented in mono-spaced bold and all distinguishable thanks to context.

Key combinations can be distinguished from an individual key by the plus sign that connects each part of a key combination. For example:

> Press **`Enter`** to execute the command.
>
> Press **`Ctrl`**+**`Alt`**+**`F2`** to switch to a virtual terminal.

The first example highlights a particular key to press. The second example highlights a key combination: a set of three keys pressed simultaneously.

If source code is discussed, class names, methods, functions, variable names and returned values mentioned within a paragraph will be presented as above, in **`mono-spaced bold`**. For example:

> File-related classes include **`filesystem`** for file systems, **`file`** for files, and **`dir`** for directories. Each class has its own associated set of permissions.

Proportional Bold

This denotes words or phrases encountered on a system, including application names; dialog-box text; labeled buttons; check-box and radio-button labels; menu titles and submenu titles. For example:

> Choose **System** → **Preferences** → **Mouse** from the main menu bar to launch **Mouse Preferences**. In the **Buttons** tab, select the **`Left-handed mouse`** check box and click **`Close`** to switch the primary mouse button from the left to the right (making the mouse suitable for use in the left hand).
>
> To insert a special character into a **`gedit`** file, choose **Applications** → **Accessories** → **Character Map** from the main menu bar. Next, choose **Search** → **Find…** from the **Character Map** menu bar, type the name of the character in the **Search** field and click **`Next`**. The character you sought will be highlighted in the

> **Character Table**. Double-click this highlighted character to place it in the **Text to copy** field and then click the **Copy** button. Now switch back to your document and choose **Edit** → **Paste** from the **gedit** menu bar.

The above text includes application names; system-wide menu names and items; application-specific menu names; and buttons and text found within a GUI interface, all presented in proportional bold and all distinguishable by context.

Mono-spaced Bold Italic or ***Proportional Bold Italic***

Whether mono-spaced bold or proportional bold, the addition of italics indicates replaceable or variable text. Italics denotes text you do not input literally or displayed text that changes depending on circumstance. For example:

> To connect to a remote machine using ssh, type **ssh *username*@*domain.name*** at a shell prompt. If the remote machine is **example.com** and your username on that machine is john, type **ssh john@example.com**.
>
> The **mount -o remount *file-system*** command remounts the named file system. For example, to remount the **/home** file system, the command is **mount -o remount /home**.
>
> To see the version of a currently installed package, use the **rpm -q *package*** command. It will return a result as follows: ***package-version-release***.

Note the words in bold italics above: username, domain.name, file-system, package, version and release. Each word is a placeholder, either for text you enter when issuing a command or for text displayed by the system.

Aside from standard usage for presenting the title of a work, italics denotes the first use of a new and important term. For example:

> Publican is a *DocBook* publishing system.

1.2. Pull-quote Conventions

Terminal output and source code listings are set off visually from the surrounding text.

Output sent to a terminal is set in **mono-spaced roman** and presented thus:

```
books        Desktop   documentation  drafts  mss    photos   stuff  svn
books_tests  Desktop1  downloads      images  notes  scripts  svgs
```

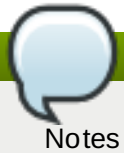
Source-code listings are also set in **mono-spaced roman** but add syntax highlighting as follows:

```
static int kvm_vm_ioctl_deassign_device(struct kvm *kvm,
                  struct kvm_assigned_pci_dev *assigned_dev)
{
         int r = 0;
         struct kvm_assigned_dev_kernel *match;

         mutex_lock(&kvm->lock);

         match = kvm_find_assigned_dev(&kvm->arch.assigned_dev_head,
                                       assigned_dev->assigned_dev_id);
         if (!match) {
                 printk(KERN_INFO "%s: device hasn't been assigned
```

```
before, "
                    "so cannot be deassigned\n", __func__);
                r = -EINVAL;
                goto out;
        }

        kvm_deassign_device(kvm, match);

        kvm_free_assigned_device(kvm, match);

out:
        mutex_unlock(&kvm->lock);
        return r;
}
```

1.3. Notes and Warnings

Finally, we use three visual styles to draw attention to information that might otherwise be overlooked.



Note

Notes are tips, shortcuts or alternative approaches to the task at hand. Ignoring a note should have no negative consequences, but you might miss out on a trick that makes your life easier.

Important

Important boxes detail things that are easily missed: configuration changes that only apply to the current session, or services that need restarting before an update will apply. Ignoring a box labeled “Important” will not cause data loss but may cause irritation and frustration.

Warning

Warnings should not be ignored. Ignoring warnings will most likely cause data loss.

2. Getting Help and Giving Feedback

2.1. Do You Need Help?

If you experience difficulty with a procedure described in this documentation, visit the Red Hat Customer Portal at http://access.redhat.com. Through the customer portal, you can:

- search or browse through a knowledgebase of technical support articles about Red Hat products.
- submit a support case to Red Hat Global Support Services (GSS).
- access other product documentation.

Red Hat also hosts a large number of electronic mailing lists for discussion of Red Hat software and technology. You can find a list of publicly available mailing lists at https://www.redhat.com/mailman/listinfo. Click on the name of any mailing list to subscribe to that list or to access the list archives.

2.2. We Need Feedback!

If you find a typographical error in this manual, or if you have thought of a way to make this manual better, we would love to hear from you! Please submit a report in Bugzilla: http://bugzilla.redhat.com/ against the product **Red Hat Enterprise Virtualization Manager**.

When submitting a bug report, be sure to mention the manual's identifier: *Guides-Technical Reference Guide*

If you have a suggestion for improving the documentation, try to be as specific as possible when describing it. If you have found an error, please include the section number and some of the surrounding text so we can find it easily.

Chapter 1. Introduction

1.1. Red Hat Enterprise Virtualization Manager

The Red Hat Enterprise Virtualization Manager provides centralized management for a virtualized environment. A number of different interfaces can be used to access the Red Hat Enterprise Virtualization Manager. Each interface facilitates access to the virtualized environment in a different manner.

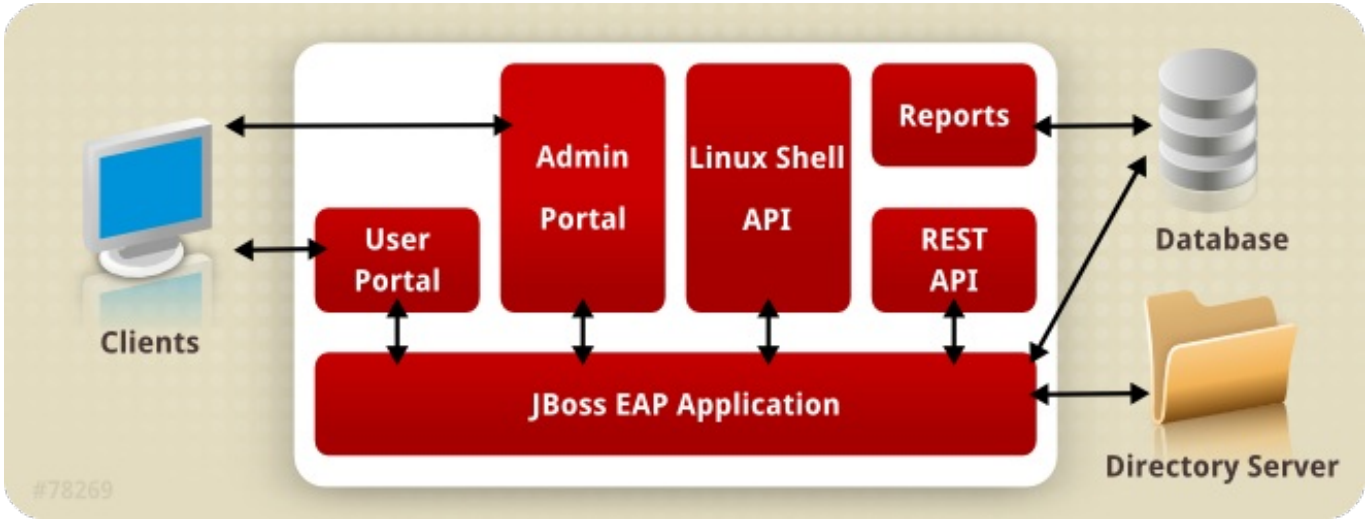


Figure 1.1. Red Hat Enterprise Virtualization Manager Architecture

The Red Hat Enterprise Virtualization Manager provides graphical interfaces and an *Application Programming Interface* (API). Each interface connects to the Manager, an application delivered by an embedded instance of the *Red Hat JBoss Enterprise Application Platform*. There are a number of other components which support the Red Hat Enterprise Virtualization Manager in addition to Red Hat JBoss Enterprise Application Platform.

Report a bug

1.2. Red Hat Virtualization Hypervisor

A Red Hat Enterprise Virtualization environment has one or more hosts attached to it. A host is a server that provides the physical hardware that virtual machines make use of.

Red Hat Enterprise Virtualization Hypervisor hosts run an optimized operating system installed using a special, customized installation media specifically for creating virtualization hosts.

Red Hat Enterprise Linux hosts are servers running a standard Red Hat Enterprise Linux operating system that has been configured after installation to permit use as a host.

Both methods of host installation result in hosts that interact with the rest of the virtualized environment in the same way, and so, will both referred to as *hosts*.

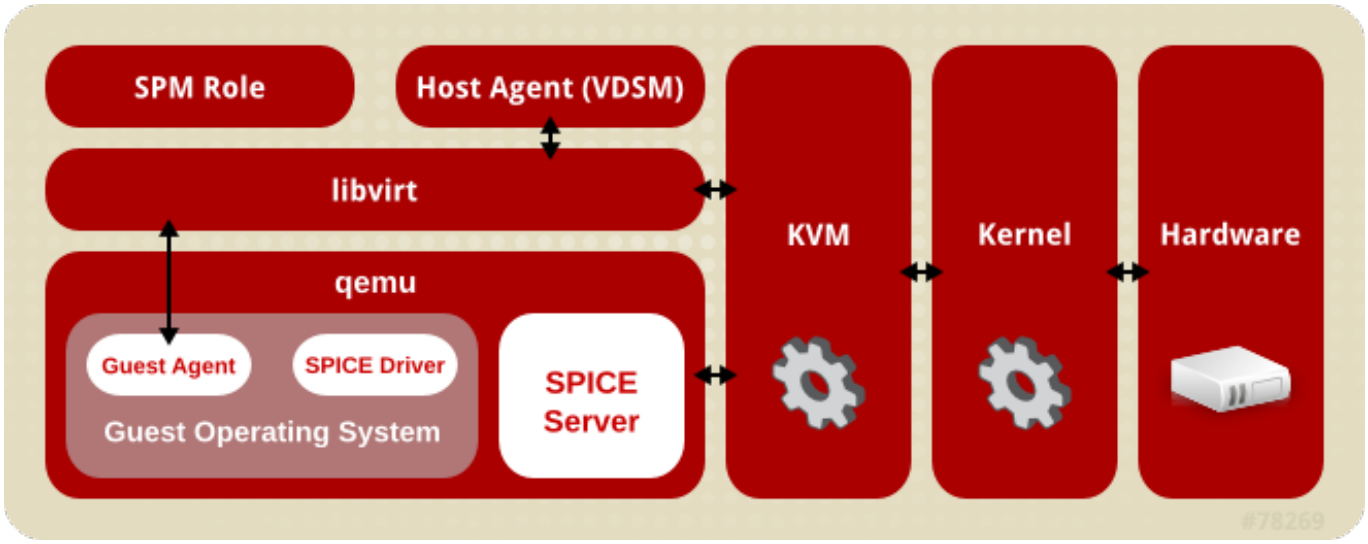


Figure 1.2. Host Architecture

Kernel-based Virtual Machine (KVM)

The Kernel-based Virtual Machine (KVM) is a loadable kernel module that provides full virtualization through the use of the Intel VT or AMD-V hardware extensions. Though KVM itself runs in kernel space, the guests running upon it run as individual *QEMU* processes in user space. KVM allows a host to make its physical hardware available to virtual machines.

QEMU

QEMU is a multi-platform emulator used to provide full system emulation. QEMU emulates a full system, for example a PC, including one or more processors, and peripherals. QEMU can be used to launch different operating systems or to debug system code. QEMU, working in conjunction with KVM and a processor with appropriate virtualization extensions, provides full hardware assisted virtualization.

Red Hat Enterprise Virtualization Manager Host Agent, VDSM

In Red Hat Enterprise Virtualization, **VDSM** initiates actions on virtual machines and storage. It also facilitates inter-host communication. VDSM monitors host resources such as memory, storage, and networking. Additionally, VDSM manages tasks such as virtual machine creation, statistics accumulation, and log collection. A VDSM instance runs on each host and receives management commands from the Red Hat Enterprise Virtualization Manager using the re-configurable port **54321**.

VDSM-REG

VDSM uses **VDSM-REG** to register each host with the Red Hat Enterprise Virtualization Manager. **VDSM-REG** supplies information about itself and its host using port **80** or port **443**.

libvirt

Libvirt facilitates the management of virtual machines and their associated virtual devices. When Red Hat Enterprise Virtualization Manager initiates virtual machine life-cycle commands (start, stop, reboot), VDSM invokes libvirt on the relevant host machines to execute them.

Storage Pool Manager, SPM

The Storage Pool Manager (SPM) is a role assigned to one host in a data center. The SPM

host has sole authority to make all storage domain structure metadata changes for the data center. This includes creation, deletion, and manipulation of virtual disk images, snapshots, and templates. It also includes allocation of storage for sparse block devices on a *Storage Area Network*(SAN). The role of SPM can be migrated to any host in a data center. As a result, all hosts in a data center must have access to all the storage domains defined in the data center.

Red Hat Enterprise Virtualization Manager ensures that the SPM is always available. In case of storage connectivity errors, the Manager re-assigns the SPM role to another host.

Guest Operating System

Guest operating systems can be installed without modification on virtual machines in a Red Hat Enterprise Virtualization environment. The guest operating system, and any applications on the guest, are unaware of the virtualized environment and run normally.

Red Hat provides enhanced device drivers that allow faster and more efficient access to virtualized devices. You can also install the Red Hat Enterprise Virtualization Guest Agent on guests, which provides enhanced guest information to the management console.

Report a bug

1.3. Features Requiring a Compatibility Upgrade to Red Hat Enterprise Virtualization 3.3

Some of the new features in Red Hat Enterprise Virtualization are only available if your data centers, clusters, and storage have a compatibility version of **3.3**.

Table 1.1. Features Requiring a Compatibility Upgrade to Red Hat Enterprise Virtualization 3.3

Feature	Description
Libvirt-to-libvirt virtual machine migration	Perform virtual machine migration using libvirt-to-libvirt communication. This is safer, more secure, and has less host configuration requirements than native KVM migration, but has a higher overhead on the host CPU.
Isolated network to carry virtual machine migration traffic	Separates virtual machine migration traffic from other traffic types, like management and display traffic. Reduces chances of migrations causing a network flood that disrupts other important traffic types.
Define a gateway per logical network	Each logical network can have a gateway defined as separate from the management network gateway. This allows more customizable network topologies.
Snapshots including RAM	Snapshots now include the state of a virtual machine's memory as well as disk.
Optimized iSCSI device driver for virtual machines	Virtual machines can now consume iSCSI storage as virtual hard disks using an optimized device driver.
Host support for MOM management of memory overcommitment	MOM is a policy-driven tool that can be used to manage overcommitment on hosts. Currently MOM supports control of memory ballooning and KSM.

Feature	Description
GlusterFS data domains.	Native support for the GlusterFS protocol was added as a way to create storage domains, allowing Gluster data centers to be created.
Custom device property support	In addition to defining custom properties of virtual machines, you can also define custom properties of virtual machine devices.
Multiple monitors using a single virtual PCI device	Drive multiple monitors using a single virtual PCI device, rather than one PCI device per monitor.
Updatable storage server connections	It is now possible to edit the storage server connection details of a storage domain.
Check virtual hard disk alignment	Check if a virtual disk, the filesystem installed on it, and its underlying storage are aligned. If it is not aligned, there may be a performance penalty.
Extendable virtual machine disk images	You can now grow your virtual machine disk image when it fills up.
OpenStack Image Service integration	Red Hat Enterprise Virtualization supports the OpenStack Image Service. You can import images from and export images to an Image Service repository.
Gluster hook support	You can manage Gluster hooks, which extend volume life cycle events, from Red Hat Enterprise Virtualization Manager.
Gluster host UUID support	This feature allows a Gluster host to be identified by the Gluster server UUID generated by Gluster in addition to identifying a Gluster host by IP address.
Network quality of service (QoS) support	Limit the inbound and outbound network traffic at the virtual NIC level.
Cloud-Init support	Cloud-Init allows you to automate early configuration tasks in your virtual machines, including setting hostnames, authorized keys, and more.

Report a bug

Chapter 2. Red Hat Enterprise Virtualization Architecture

2.1. Interfaces for Accessing the Manager

User Portal

Desktop virtualization provides users with a desktop environment that is similar a personal computer's desktop environment. The User Portal is for delivering *Virtual Desktop Infrastructure* to users. Users access the User Portal through a web browser to display and access their assigned virtual desktops. The actions available to a user in the User Portal are set by a system administrator. Standard users can start, stop, and use desktops that are assigned to them by the system administrator. Power users can perform some administrative actions. Both types of user access the User Portal from the same URL, and are presented with options appropriate to their permission level on login.

Standard User Access

Standard users are able to power their virtual desktops on and off and connect to them through the User Portal. Direct connection to virtual machines is facilitated with *Simple Protocol for Independent Computing Environments (SPICE)* or *Virtual Network Computing (VNC)* clients. Both protocols provide the user with an environment similar to a locally installed desktop environment. The administrator specifies the protocol used to connect to a virtual machine at the time of the virtual machine's creation.

More information on the actions available from the User Portal as well as supported browsers and clients can be found in the *User Portal Guide*.

Power User Access

The Red Hat Enterprise Virtualization User Portal provides power users with a graphical user interface to create, use, and monitor virtual resources. System administrators can delegate some administration tasks by granting users power user access. In addition to the tasks that can be performed by standard users, power users can:

- Create, edit, and remove virtual machines.
- Manage virtual disks and network interfaces.
- Assign user permissions to virtual machines.
- Create and use templates to rapidly deploy virtual machines.
- Monitor resource usage and high-severity events.
- Create and use snapshots to restore virtual machines to previous states.

Power users can perform virtual machine administration tasks to be delegated. Data center and cluster level administration tasks are saved for the environment administrator.

Administration Portal

The Administration Portal is the graphical administration interface of the Red Hat Enterprise Virtualization Manager server. Using it administrators can monitor, create, and maintain all elements of the virtualized environment using from web browsers. Tasks which can be performed from the Administration Portal include:

- Creation and management of virtual infrastructure (networks, storage domains).
- Installation and management of hosts.
- Creation and management of logical entities (data centers, clusters).
- Creation and management of virtual machines.
- Red Hat Enterprise Virtualization user and permission management.

The Administration Portal is displayed using the JavaScript.

Administration Portal functions are discussed in further detail in the *Red Hat Enterprise Virtualization Administration Guide*. Information on the browsers and platforms that are supported by the Administration Portal can be found in the *Red Hat Enterprise Virtualization Installation Guide*.

Representational State Transfer (REST) API

The Red Hat Enterprise Virtualization REST API provides a software interface for the interrogation and control of the Red Hat Enterprise Virtualization environment. The REST API can be used by any programming language that supports HTTP actions.

Using the REST API developers and administrators can:

- Integrate with enterprise IT systems.
- Integrate with third party virtualization software.
- Perform automated maintenance and error checking tasks.
- Use scripts to automate repetitive tasks in a Red Hat Enterprise Virtualization environment.

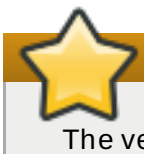
See the *REST API Guide* for the API specification and usage examples.

Report a bug

2.2. Components that Support the Manager

Red Hat JBoss Enterprise Application Platform

Red Hat JBoss Enterprise Application Platform is a Java application server. It provides a framework to support efficient development and delivery of cross-platform Java applications. The Red Hat Enterprise Virtualization Manager is delivered using Red Hat JBoss Enterprise Application Platform.



Important

The version of the Red Hat JBoss Enterprise Application Platform bundled with Red Hat Enterprise Virtualization Manager is **not** to be used to serve other applications. It has been customized for the specific purpose of serving the Red Hat Enterprise Virtualization Manager. Using the Red Hat JBoss Enterprise Application Platform that is included with the Manager for additional purposes adversely affects its ability to service the Red Hat Enterprise Virtualization environment.

Gathering Reports and Historical Data

The Red Hat Enterprise Virtualization Manager includes a data warehouse that collects monitoring data about hosts, virtual machines, and storage. A number of pre-defined reports are available. Customers can analyze their environments and create reports using any query tools that support SQL.

The Red Hat Enterprise Virtualization Manager installation process creates two databases. These databases are created on a Postgres instance which is selected during installation.

- The engine database is the primary data store used by the Red Hat Enterprise Virtualization Manager. Information about the virtualization environment like its state, configuration, and performance are stored in this database.
- The ovirt_engine_history database contains configuration information and statistical metrics which are collated over time from the engine operational database. The configuration data in the engine database is examined every minute, and changes are replicated to the ovirt_engine_history database. Tracking the changes to the database provides information on the objects in the database. This enables you to analyze and enhance the performance of your Red Hat Enterprise Virtualization environment and resolve difficulties.

  For more information on generating reports based on the ovirt_engine_history database see the *Red Hat Enterprise Virtualization Administration Guide*.

Important

The replication of data in the ovirt_engine_history database is performed by the **RHEVM History Service**, ovirt-engine-dwhd.

Directory services

Directory services provide centralized network-based storage of user and organizational information. Types of information stored include application settings, user profiles, group data, policies, and access control. The Red Hat Enterprise Virtualization Manager supports Active Directory, Identity Management (IdM), OpenLDAP, and Red Hat Directory Server 9. There is also a local, internal domain for administration purposes only. This internal domain has only one user: the admin user.

Report a bug

2.3. Storage

Red Hat Enterprise Virtualization uses a centralized storage system for virtual machine disk images, templates, snapshots, and ISO files. Storage is logically grouped into storage pools, which are comprised of storage domains. A storage domain is a combination of storage capacity and metadata that describes the internal structure of the storage. There are three types of storage domain; data, export, and ISO.

The data storage domain is the only one required by each data center. A data storage domain is exclusive to a single data center. Export and ISO domains are optional. Storage domains are shared resources, and must be accessible to all hosts in a data center.

Storage networking can be implemented using Network File System (NFS), Internet Small Computer System Interface (iSCSI), or Fibre Channel Protocol (FCP). Red Hat Enterprise Virtualization additionally provides the *unsupported* ability to use any POSIX compliant networked filesystem as a data domain.

On NFS (and other POSIX compliant filesystems) domains, all virtual disks, templates, and snapshots are simple files.

On SAN (iSCSI/FCP) domains, block devices are aggregated by by Logical Volume Manager (LVM)into a Volume Group (VG). Each virtual disk, template and snapshot is a Logical Volume (LV) on the VG. See *Red Hat Enterprise Linux Logical Volume Manager Administration Guide* for more information on LVM.

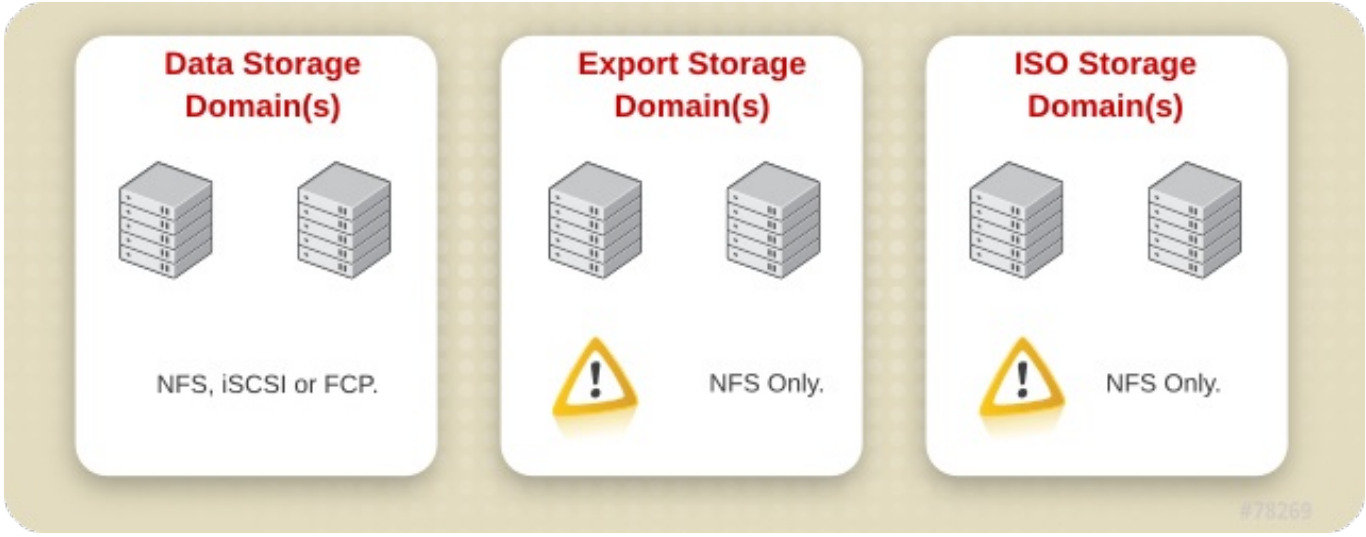


Figure 2.1. Storage Architecture

Data storage domain

Data domains hold the virtual hard disk images of all the virtual machines running in the environment. Templates and snapshots of the virtual machines are also stored in the data domain. A data domain cannot be shared across data centers, and the data domain must be of the same type as the data center. For example, a data center of iSCSI type must have an iSCSI data domain.

Export storage domain

An export domain is a temporary storage repository that is used to copy and move images between data centers and Red Hat Enterprise Virtualization environments. The export domain can be used to back up virtual machines and templates. An export domain can be moved between data centers, but can only be active in one data center at a time.

ISO storage domain

ISO domains store ISO files, which are logical CD-ROMs used to install operating systems and applications for the virtual machines. As a logical entity that replaces a library of physical CD-ROMs or DVDs, an ISO domain removes the data center's need for physical media. An ISO domain can be shared across different data centers.

Report a bug

2.4. Network

The Red Hat Enterprise Virtualization network architecture facilitates connectivity between the different elements of the Red Hat Enterprise Virtualization environment. The network architecture not only supports network connectivity, it also allows for network segregation.

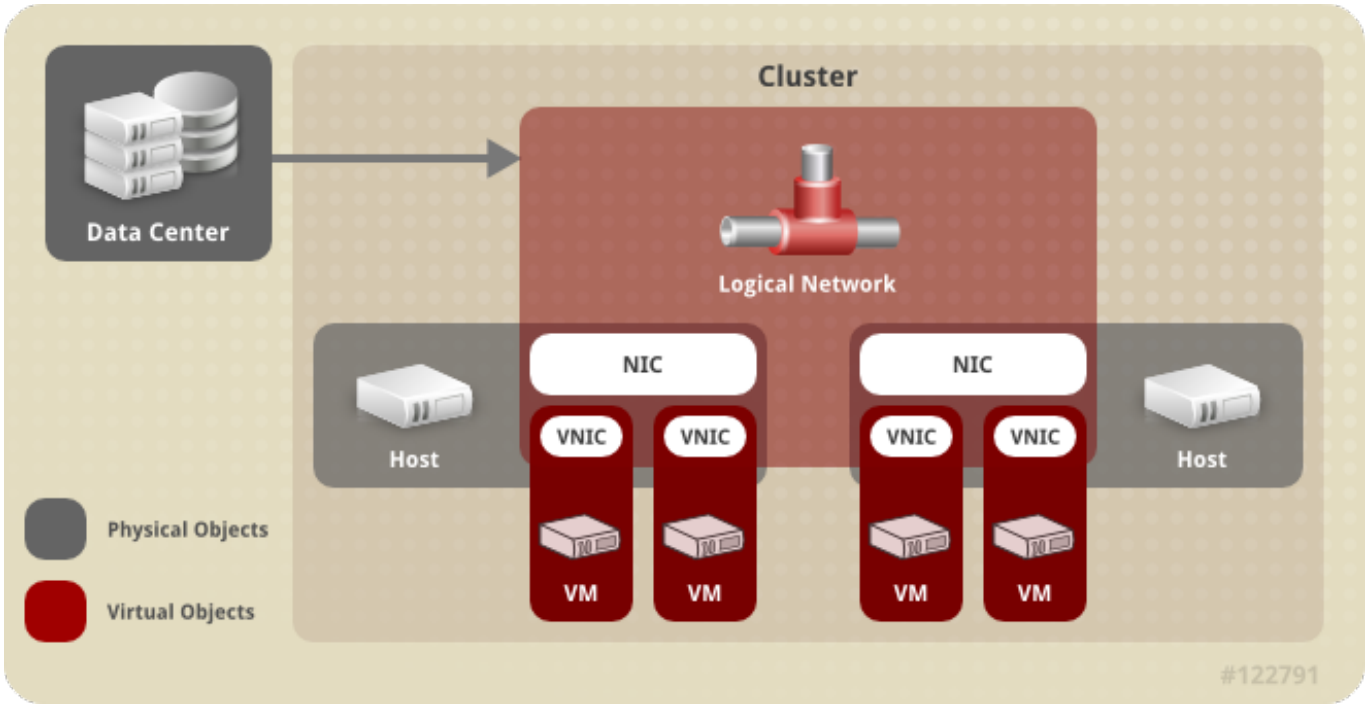


Figure 2.2. Network Architecture

Networking is defined in Red Hat Enterprise Virtualization in several layers. The underlying physical networking infrastructure must be in place and configured to allow connectivity between the hardware and the logical components of the Red Hat Enterprise Virtualization environment.

Networking Infrastructure Layer

The Red Hat Enterprise Virtualization network architecture relies on some common hardware and software devices:

- *Network Interface Controllers* (NICs) are physical network interface devices that connect a host to the network.
- *Virtual NICs* (VNICs) are logical NICs that operate using the host's physical NICs. They provide network connectivity to virtual machines.
- *Bonds* bind multiple NICs into a single interface.
- *Bridges* are a packet-forwarding technique for packet-switching networks. They form the basis of *virtual machine logical networks*.

Logical Networks

Logical networks allow segregation of network traffic based on environment requirements. The types of logical network are:

- logical networks that carry virtual machine network traffic,
- logical networks that do not carry virtual machine network traffic,
- optional logical networks,
- and required networks.

All logical networks can either be required or optional.

A logical network that carries virtual machine network traffic is implemented at the host level as a software bridge device. By default, one logical network is defined during the installation of the Red Hat Enterprise Virtualization Manager: the **rhevm** Management network.

Other logical networks that can be added by an administrator are: a dedicated storage logical network, and a dedicated display logical network. Logical networks that do not carry virtual machine traffic do not have an associated bridge device on hosts. They are associated with host network interfaces directly.

Red Hat Enterprise Virtualization 3.3 segregates management-related network traffic from migration-related network traffic. This makes it possible to use a dedicated network (without routing) for live migration, and ensures that the management network (rhevm) does not lose its connection to hypervisors during migrations.

Explanation of logical networks on different layers

Logical networks have different implications for each layer of the virtualization environment.

Data Center Layer

Logical networks are defined at the data center level. Each data center has the **rhevm** management network by default. Further logical networks are optional but recommended. Designation as a **VM Network** and a custom MTU can be set at the data center level. A logical network that is defined for a data center must also be added to the clusters that use the logical network.

Cluster Layer

Logical networks are made available from a data center, and must be added to the clusters that will use them. Each cluster is connected to the management network by default. You can optionally add to a cluster logical networks that have been defined for the cluster's parent data center. When a required logical network has been added to a cluster, it must be implemented for each host in the cluster. Optional logical networks can be added to hosts as needed.

Host Layer

Virtual machine logical networks are implemented for each host in a cluster as a software bridge device associated with a given network interface. Non-virtual machine logical networks do not have associated bridges, and are associated with host network interfaces directly. Each host has the management network implemented as a bridge using one of its network devices as a result of being included in a Red Hat Enterprise Virtualization environment. Further required logical networks that have been added to a cluster must be associated with network interfaces on each host to become operational for the cluster.

Virtual Machine Layer

Logical networks can be made available to virtual machines in the same way that a network can be made available to a physical machine. A virtual machine can have its virtual NIC connected to any virtual machine logical network that has been implemented on the host that runs it. The virtual machine then gains connectivity to any other devices or destinations that are available on the logical network it is connected to.

> **Example 2.1. Management Network**
>
> The management logical network, named **rhevm**, is created automatically when the Red

> Hat Enterprise Virtualization Manager is installed. The **rhevm** network is dedicated to management traffic between the Red Hat Enterprise Virtualization Manager and hosts. If no other specifically-purposed bridges are set up, **rhevm** is the default bridge for all traffic.

Report a bug

Chapter 3. Storage architecture

3.1. Storage Concepts

3.1.1. Data Centers

A data center is the highest level of abstraction in Red Hat Enterprise Virtualization. A data center is a container that is comprised of three types of sub-containers:

- The *storage container* holds information about storage types and storage domains, including connectivity information for storage domains. Storage is defined for a data center, and available to all clusters in the data center. All host clusters within a data center have access to the same storage domains.
- The *network container* holds information about the data center's logical networks. This includes details such as network addresses, VLAN tags and STP support. Logical networks are defined for a data center, and are optionally implemented at the cluster level.
- The *cluster container* holds clusters. Clusters are groups of hosts with compatible processor cores, either AMD or Intel processors. Clusters are migration domains; virtual machines can be live-migrated to any host within a cluster, and not to other clusters. One data center can hold multiple clusters, and each cluster can contain multiple hosts.

Report a bug

3.1.2. Storage Domains Overview

A storage domain is a centrally accessed repository for disk images and metadata. Storage domains keep virtual machine data and metadata, ISO images, snapshots, and other data in a central location that is accessible to all hosts in a data center. Storage domains also hold metadata about themselves that is used to keep virtual machine data from being corrupted.

File based storage and block based storage can be used to create storage domains. Storage domains are implemented using the Network File System (NFS) or Fibre Channel Protocol (FCP). FCP includes storage accessed using iSCSI, FCoE, and SAS.

Red Hat Enterprise Virtualization 3.3 introduces support for parallel NFS (pNFS). NFSv4.1 extends the capabilities of its predecessors by supporting high-speed I/O to clustered servers. This makes parallel I/O possible and increases scalability and overall performance. Metadata (directory information, for instance) can be stored on different servers than data is stored on; this keeps the metadata out of the data path. Metadata servers provide clients' layout information, allowing the clients to access data servers separately and in parallel. A hypervisor that uses pNFS to access a virtual machine data store can take advantage of multiple connections per session, and those connections can be distributed across multiple interfaces. pNFS supports blocks, objects, and files. Red Hat added pNFS client as a tech preview in Red Hat Enterprise Linux 6.2, and pNFS is supported out-of-the-box in Red Hat Enterprise Linux 6.4.

There are three types of storage domain: data storage domains, ISO storage domains, and export storage domains.

Report a bug

3.1.3. Types of Storage Backing Storage Domains

Storage domains can be implemented using block based and file based storage.

File Based Storage

The file based storage types supported by Red Hat Enterprise Virtualization are NFS and storage local to hosts.

Note

It is possible to use any POSIX compliant networked filesystem as a storage domain, none but NFS are supported.

File based storage is managed externally to the Red Hat Enterprise Virtualization environment.

NFS storage is managed by a Red Hat Enterprise Linux NFS server, or other third party network attached storage server.

Red Hat Enterprise Virtualization hosts can manage their own local storage file systems.

Block Based Storage

Block storage uses un-formatted block devices. Block devices are aggregated into *volume groups* by the Logical Volume Manager (LVM). An instance of LVM runs on all hosts, unaware of the instances running on other hosts. VDSM adds clustering logic on top of LVM by scanning volume groups for changes. When changes are detected, VDSM updates individual hosts by telling them to refresh their volume group information. The hosts divide the volume group into logical volumes, writing logical volume metadata to disk. If more storage capacity is added to an existing storage domain, the Red Hat Enterprise Virtualization Manager causes VDSM on each host to refresh volume group information.

A Logical Unit Number (LUN) is an individual block device. One of the supported block storage protocols, iSCSI, FCoE, or SAS, is used to connect to a LUN. The Red Hat Enterprise Virtualization Manager manages software iSCSI connections to the LUNs. All other block storage connections are managed externally to the Red Hat Enterprise Virtualization environment. Any changes in a block based storage environment, such as the creation of logical volumes, extension or deletion of logical volumes and the addition of a new LUN are handled by LVM on a specially selected host called the Storage Pool Manager. Changes are then synced by VDSM which storage metadata refreshes across all hosts in the cluster.

Report a bug

3.1.4. Storage Domain Types

Red Hat Enterprise Virtualization supports these types of storage domains, and the storage types that each of the storage domains supports:

- The *Data Storage Domain* stores the hard disk images of all virtual machines in the Red Hat Enterprise Virtualization environment. Disk images may contain an installed operating system or data stored or generated by a virtual machine. Data storage domains support NFS, iSCSI, FCP, and POSIX compliant storage. A data domain cannot be shared between multiple data centers. Additionally, it is required that the data center and data storage domain use the same protocol (for example, both must be iSCSI based).
- The *Export Storage Domain* provides transitory storage for hard disk images and virtual machine templates being transferred between data centers. Additionally, export storage domains store backed up copies of virtual machines. Export storage domains support NFS storage. Multiple

data centers can access a single export storage domain but only one data center can use it at a time.

- The *ISO Storage Domain* stores ISO files, also called images. ISO files are representations of physical CDs or DVDs. In the Red Hat Enterprise Virtualization environment the common types of ISO files are operating system installation disks, application installation disks, and guest agent installation disks. These images can be attached to virtual machines and booted in the same way that physical disks are inserted into a disk drive and booted. ISO storage domains allow all hosts within the data center to share ISOs, eliminating the need for physical optical media.

Report a bug

3.1.5. Storage Formats for Virtual Machine Disk Images

QCOW2 Formatted Virtual Machine Storage

QCOW2 is a storage format for virtual machine disk images. QCOW stands for *QEMU copy on write*. The QCOW2 format decouples the physical storage layer from the virtual layer by adding a mapping between logical and physical blocks. Each logical block is mapped to its physical offset, which enables storage over-commitment and virtual machine snapshots, where each QCOW volume only represents changes made to an underlying disk image.

The initial mapping points all logical blocks to the offsets in the backing file or volume. When a virtual machine writes data to a QCOW2 volume after a snapshot, the relevant block is read from the backing volume, modified with the new information and written into a new snapshot QCOW2 volume. Then the map is updated to point to the new place.

RAW

The RAW storage format has a performance advantage over QCOW2 in that no formatting is applied to virtual machine disk images stored in the RAW format. Virtual machine data operations on disk images stored in RAW format require no additional work from hosts. When a virtual machine writes data to a given offset in its virtual disk, the I/O is written to the same offset on the backing file or logical volume.

Raw format requires that the entire space of the defined image be preallocated unless using externally managed thin provisioned LUNs from a storage array.

Report a bug

3.1.6. Virtual Machine Disk Image Storage Allocation Policies

Preallocated Storage

All of the storage required for a virtual machine disk image is allocated prior to virtual machine creation. If a 20GB disk image is created for a virtual machine, the disk image uses 20GB of storage domain capacity. Preallocated disk images cannot be enlarged. Preallocating storage can mean faster write times because no storage allocation takes place during runtime, at the cost of flexibility. Allocating storage this way reduces the capacity of the Red Hat Enterprise Virtualization Manager to over-commit storage. Preallocated storage is recommended for virtual machines used for high intensity I/O tasks with less tolerance for latency in storage. Generally, server virtual machines fit this description.

Note

If thin provisioning functionality provided by your storage back-end is being used, preallocated storage should still be selected from the Administration Portal when provisioning storage for virtual machines.

Sparsely Allocated Storage

The upper size limit for a virtual machine disk image is set at virtual machine creation time. Initially, the disk image does not use any storage domain capacity. Usage grows as the virtual machine writes data to disk, until the upper limit is reached. Capacity is not returned to the storage domain when data in the disk image is removed. Sparsely allocated storage is appropriate for virtual machines with low or medium intensity I/O tasks with some tolerance for latency in storage. Generally, desktop virtual machines fit this description.

Note

If thin provisioning functionality is provided by your storage back-end, it should be used as the preferred implementation of thin provisioning. Storage should be provisioned from the graphical user interface as preallocated, leaving thin provisioning to the back-end solution.

Report a bug

3.1.7. Storage Domain Autorecovery in Red Hat Enterprise Virtualization

Hosts in a Red Hat Enterprise Virtualization environment monitor storage domains in their data centers by reading metadata from each domain. A storage domain becomes inactive when all hosts in a data center report that they cannot access the storage domain.

Prior to Red Hat Enterprise Virtualization 3.1, storage domains that became inactive were disconnected by the Manager. Reconnecting to storage when connection issues had been resolved required manual administrator intervention.

Red Hat Enterprise Virtualization 3.1 introduced *storage domain autorecovery*. Rather than disconnecting an inactive storage domain, the Manager now assumes that the storage domain has become inactive temporarily, because of a temporary network outage for example. Once every 5 minutes, the Manager attempts to re-activate any inactive storage domains.

Administrator intervention may be required to remedy the cause of the storage connectivity interruption, but the Manager handles re-activating storage domains as connectivity is restored.

Report a bug

3.1.8. The Storage Pool Manager

Red Hat Enterprise Virtualization uses metadata to describe the internal structure of storage domains. Structural metadata is written to a segment of each storage domain. Hosts work with the storage domain metadata based on a single writer, and multiple readers configuration. Storage domain structural metadata tracks image and snapshot creation and deletion, and volume and domain extension.

The Red Hat Enterprise Virtualization host that can make changes to the structure of the data domain

is known as the Storage Pool Manager (SPM). The SPM coordinates all metadata changes in the data center, such as creating and deleting disk images, creating and merging snapshots, copying images between storage domains, creating templates and storage allocation for block devices. There is one SPM for every data center. All other hosts can only read storage domain structural metadata.

A host can be manually selected as the SPM, or it can be assigned by the Red Hat Enterprise Virtualization Manager. The Manager assigns the SPM role by causing a potential SPM host to attempt to assume a *storage-centric lease*. The lease allows the SPM host to write storage metadata. It is storage-centric because it is written to the storage domain rather than being tracked by the Manager or hosts. Storage-centric leases are written to a special logical volume in the master storage domain called **`leases`**. Metadata about the structure of the data domain is written to a special logical volume called **`metadata`**. Changes to the **`metadata`** logical volume are protected against by the **`leases`** logical volume.

The Manager uses VDSM to issue the **`spmStart`** command to a host, causing VDSM on that host to attempt to assume the storage-centric lease. If the host is successful it becomes the SPM and retains the storage-centric lease until the Red Hat Enterprise Virtualization Manager requests that a new host assume the role of SPM.

The Manager moves the SPM role to another host if:

- the SPM host can not access all storage domains, but can access the master storage domain.
- the SPM host is unable to renew the lease because of a loss of storage connectivity or the lease volume is full and no write operation can be performed.
- the SPM host crashes.

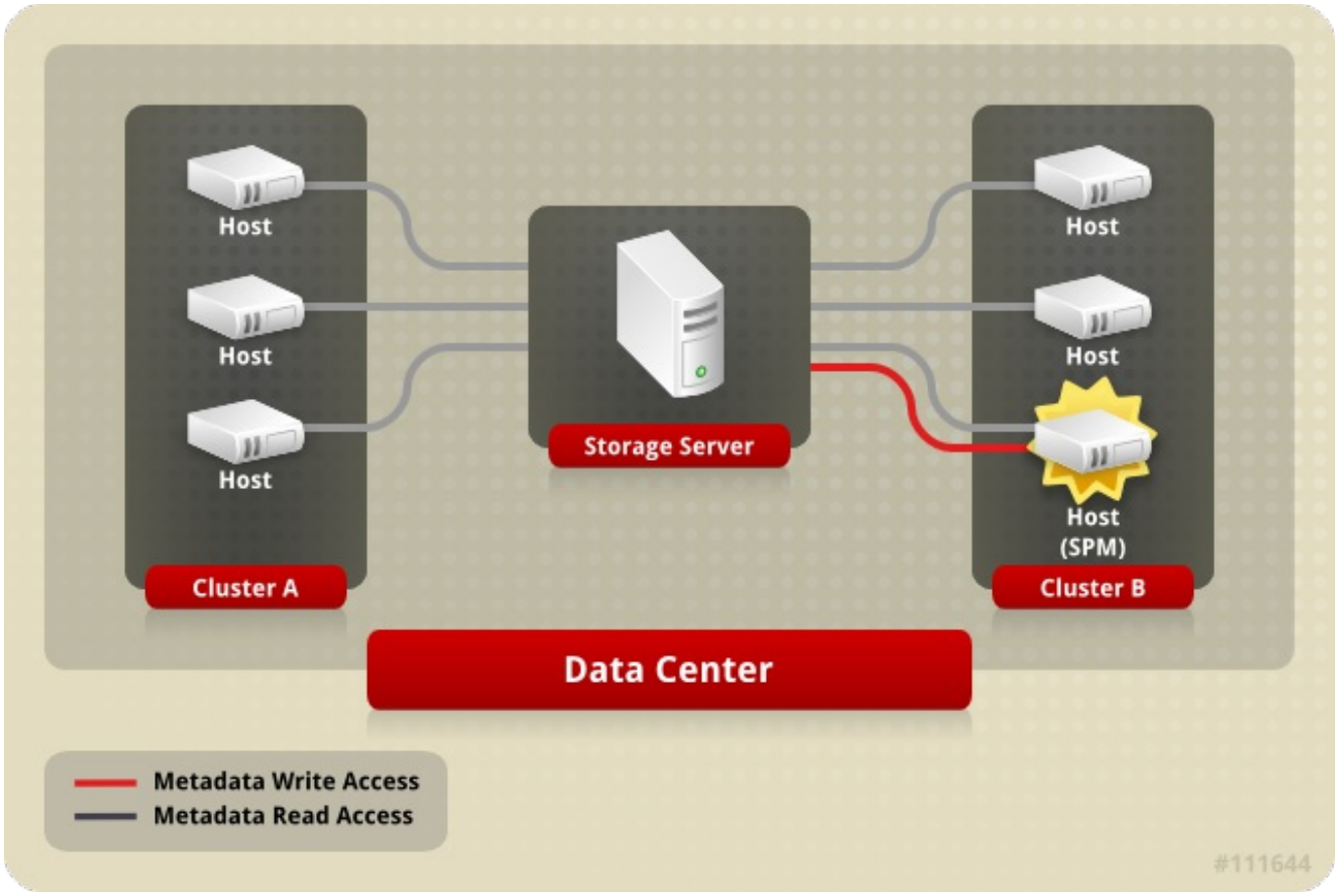


Figure 3.1. The Storage Pool Manager Exclusively Writes Structural Metadata.

Report a bug

3.1.9. Storage Pool Manager Selection Process

If a host has not been manually assigned the Storage Pool Manager (SPM) role, the SPM selection process is initiated and managed by the Red Hat Enterprise Virtualization Manager.

First, the Red Hat Enterprise Virtualization Manager requests that VDSM confirm which host has the storage-centric lease.

The Red Hat Enterprise Virtualization Manager tracks the history of SPM assignment from the initial creation of a storage domain onward. The availability of the SPM role is confirmed in three ways:

- The "getSPMstatus" command: the Manager uses VDSM to check with the host that had SPM status last and receives one of "SPM", "Contending", or "Free".
- The metadata volume for a storage domain contains the last host with SPM status.
- The metadata volume for a storage domain contains the version of the last host with SPM status.

If an operational, responsive host retains the storage-centric lease, the Red Hat Enterprise Virtualization Manager marks that host SPM in the administrator portal. No further action is taken.

If the SPM host does not respond, it is considered unreachable. If power management has been configured for the host, it is automatically fenced. If not, it requires manual fencing. The Storage Pool Manager role cannot be assigned to a new host until the previous Storage Pool Manager is fenced.

When the Storage Pool Manager role and storage-centric lease are free, the Red Hat Enterprise Virtualization Manager assigns them to a randomly selected operational host in the data center.

If the Storage Pool Manager role assignment fails on a new host, the Red Hat Enterprise Virtualization Manager adds the host to a list containing the hosts the operation has failed on. On subsequent iterations of the SPM selection, the Red Hat Enterprise Virtualization Manager attempts to assign the role to a host that is not included in the list.

The Red Hat Enterprise Virtualization Manager continues request that the Storage Pool Manager role and storage-centric lease be assumed by a randomly selected host that is not on the list of failed hosts until the SPM selection succeeds.

Each time the current SPM is unresponsive or unable to fulfill its responsibilities, the Red Hat Enterprise Virtualization Manager initiates the Storage Pool Manager selection process.

Report a bug

3.1.10. Exclusive Resources and Sanlock in Red Hat Enterprise Virtualization

Certain resources in the Red Hat Enterprise Virtualization environment must be accessed exclusively.

The SPM role is one such resource. If more than one host were to become the SPM, there would be a risk of data corruption as the same data could be changed from two places at once.

Prior to Red Hat Enterprise Virtualization 3.1, SPM exclusivity was maintained and tracked using a VDSM feature called *safelease*. The lease was written to a special area on all of the storage domains in a data center. All of the hosts in an environment could track SPM status in a network-independent way. The VDSM's safe lease only maintained exclusivity of one resource: the SPM role.

Sanlock provides the same functionality, but treats the SPM role as one of the resources that can be locked. Sanlock is more flexible because it allows additional resources to be locked.

Applications that require resource locking can register with Sanlock. Registered applications can then request that Sanlock lock a resource on their behalf, so that no other application can access it. For example, instead of VDSM locking the SPM status, VDSM now requests that Sanlock do so.

Locks are tracked on disk in a *lockspace*. There is one lockspace for every storage domain. In the case of the lock on the SPM resource, each host's liveness is tracked in the lockspace by the host's ability to renew the hostid it received from the Manager when it connected to storage, and to write a timestamp to the lockspace at a regular interval. The *ids* logical volume tracks the unique identifiers of each host, and is updated every time a host renews its hostid. The SPM resource can only be held by a live host.

Resources are tracked on disk in the *leases* logical volume. A resource is said to be *taken* when its representation on disk has been updated with the unique identifier of the process that has taken it. In the case of the SPM role, the SPM resource is updated with the hostid that has taken it.

The Sanlock process on each host only needs to check the resources once to see that they are taken. After an initial check, Sanlock can monitor the lockspaces until timestamp of the host with a locked resource becomes stale.

Sanlock monitors the applications that use resources. For example, VDSM is monitored for SPM status and hostid. If the host is unable to renew it's hostid from the Manager, it loses exclusivity on all resources in the lockspace. Sanlock updates the resource to show that it is no longer taken.

If the SPM host is unable to write a timestamp to the lockspace on the storage domain for a given amount of time, the host's instance of Sanlock requests that the VDSM process release its resources. If the VDSM process responds, its resources are released, and the SPM resource in the lockspace can be taken by another host.

If VDSM on the SPM host does not respond to requests to release resources, Sanlock on the host kills the VDSM process. If the kill command is unsuccessful, Sanlock escalates by attempting to kill VDSM using sigkill. If the sigkill is unsuccessful, Sanlock depends on the *watchdog daemon* to reboot the host.

Every time VDSM on the host renews its hostid and writes a timestamp to the lockspace, the watchdog daemon receives a *pet*. When VDSM is unable to do so, the watchdog daemon is no longer being petted. After the watchdog daemon has not received a pet for a given amount of time, it reboots the host. This final level of escalation, if reached, guarantees that the SPM resource is released, and can be taken by another host.

Report a bug

3.2. Block based storage provisioning mechanics

3.2.1. Thin Provisioning and Storage Over-Commitment

The Red Hat Enterprise Virtualization Manager provides provisioning policies to optimize storage usage within the virtualization environment. A thin provisioning policy allows you to over-commit storage resources, provisioning storage based on the actual storage usage of your virtualization environment.

Storage over-commitment is the allocation of more storage to virtual machines than is physically available in the storage pool. Generally, virtual machines use less storage than what has been allocated to them. Thin provisioning allows a virtual machine to operate as if the storage defined for it has been completely allocated, when in fact only a fraction of the storage has been allocated.

Note

While the Red Hat Enterprise Virtualization Manager provides its own thin provisioning function, you should use the thin provisioning functionality of your storage back-end if it provides one.

To support storage over-commitment, a threshold is defined in VDSM which compares logical storage allocation with actual storage usage. This threshold is used to make sure that the data written to a disk image is smaller than the logical volume that backs it. QEMU identifies the highest offset written to in a logical volume, which indicates the point of greatest storage use. VDSM monitors the highest offset marked by QEMU to ensure that the usage does not cross the defined threshold. So long as VDSM continues to indicate that the highest offset remains below the threshold, the Red Hat Enterprise Virtualization Manager knows that the logical volume in question has sufficient storage to continue operations.

When QEMU indicates that usage has risen to exceed the threshold limit, VDSM communicates to the Manager that the disk image will soon reach the size of it's logical volume. The Red Hat Enterprise Virtualization Manager requests that the SPM host extend the logical volume. This process can be repeated as long as the data storage domain for the data center has available space. When the data storage domain runs out of available free space, you must manually add storage capacity to expand it.

Report a bug

3.2.2. Logical Volume Extension

The Red Hat Enterprise Virtualization Manager uses thin provisioning to over-commit the storage available in a storage pool, and allocates more storage than is physically available. Virtual machines write data as they operate. A virtual machine with a thinly-provisioned disk image will eventually write more data than the logical volume backing it's disk image can hold. When this happens, logical volume extension is used to provide additional storage and facilitate the continued operations for the virtual machine.

Red Hat Enterprise Virtualization provides a thin provisioning mechanism over LVM. When using QCOW2 formatted storage, Red Hat Enterprise Virtualization relies on the host system process *qemu-kvm* to map storage blocks on disk to logical blocks in a sequential manner. This allows, for example, the definition of a logical 100GB disk backed by a 1GB logical volume. When qemu-kvm crosses a usage threshold set by VDSM, the local VDSM instance makes a request to the SPM for the logical volume to be extended by another one gigabyte. VDSM on the host running a virtual machine in need of volume extension notifies the SPM VDSM that more space is required. The SPM extends the logical volume and the SPM VDSM instance causes the host VDSM to refresh volume group information and recognize that the extend operation is complete. The host can continue operations.

Logical Volume extension does not require that a host know which other host is the SPM; it could even be the SPM itself. The storage extension communication is done via a storage mailbox. The storage mailbox is a dedicated logical volume on the data storage domain. A host that needs the SPM to extend a logical volume writes a message in an area designated to that particular host in the storage mailbox. The SPM periodically reads the incoming mail, performs requested logical volume extensions, and writes a reply in the outgoing mail. After sending the request, a host monitors its incoming mail for responses every two seconds. When the host receives a successful reply to its logical volume extension request, it refreshes the logical volume map in device mapper to recognize the newly allocated storage.

When the physical storage available to a storage pool is nearly exhausted, multiple images can run out of usable storage with no means to replenish their resources. A storage pool that exhausts its

storage causes QEMU to return an **enospc error**, which indicates that the device no longer has any storage available. At this point, running virtual machines are automatically paused and manual intervention is required to add a new LUN to the volume group.

When a new LUN is added to the volume group, the Storage Pool Manager automatically distributes the additional storage to logical volumes that need it. The automatic allocation of additional resources allows the relevant virtual machines to automatically continue operations uninterrupted or resume operations if stopped.

Report a bug

Chapter 4. Network architecture

4.1. Network Architecture

Red Hat Enterprise Virtualization networking can be discussed in terms of basic networking, networking within a cluster, and host networking configurations. Basic networking terms cover the basic hardware and software elements that facilitate networking. Networking within a cluster includes network interactions among cluster level objects such as hosts, logical networks and virtual machines. Host networking configurations covers supported configurations for networking within a host.

A well designed and built network ensures, for example, that high bandwidth tasks receive adequate bandwidth, that user interactions are not crippled by latency, and virtual machines can be successfully migrated within a migration domain. A poorly built network can cause, for example, unacceptable latency, and migration and cloning failures resulting from network flooding.

Report a bug

4.2. Basic networking

4.2.1. Introduction: Basic Networking Terms

Red Hat Enterprise Virtualization provides networking functionality between virtual machines, virtualization hosts, and wider networks using:

- A Network Interface Controller (NIC)
- A Bridge
- A Bond
- A Virtual NIC
- A Virtual LAN (VLAN)

NICs, bridges, and VNICs allow for network communication between hosts, virtual machines, local area networks, and the Internet. Bonds and VLANs are optionally implemented to enhance security, fault tolerance, and network capacity.

Report a bug

4.2.2. Network Interface Controller (NIC)

The *NIC (Network Interface Controller)* is a network adapter or LAN adapter that connects a computer to a computer network. The NIC operates on both the physical and data link layers of the machine and allows network connectivity. All virtualization hosts in a Red Hat Enterprise Virtualization environment have at least one NIC, though it is more common for a host to have two or more NICs.

One physical NIC can have multiple Virtual NICs (VNICs) logically connected to it. A virtual NIC acts as a physical network interface for a virtual machine. To distinguish between a VNIC and the NIC that supports it, the Red Hat Enterprise Virtualization Manager assigns each VNIC a unique MAC address.

Report a bug

4.2.3. Bridge

A *Bridge* is a software device that uses packet forwarding in a packet-switched network. Bridging allows multiple network interface devices to share the connectivity of one NIC and appear on a network as separate physical devices. The bridge examines a packet's source addresses to determine relevant target addresses. Once the target address is determined, the bridge adds the location to a table for future reference. This allows a host to redirect network traffic to virtual machine associated VNICs that are members of a bridge.

In Red Hat Enterprise Virtualization a logical network is implemented using a bridge. It is the bridge rather than the physical interface on a host that receives an IP address. The IP address associated with the bridge is not required to be within the same subnet as the virtual machines that use the bridge for connectivity. If the bridge is assigned an IP address on the same subnet as the virtual machines that use it, the host is addressable within the logical network by virtual machines. As a rule it is not recommended to run network exposed services on a Red Hat Enterprise Virtualization host. Guests are connected to a logical network by their VNICs, and the host is connected to remote elements of the logical network using its NIC. Each guest can have the IP address of its VNIC set independently, by DHCP or statically. Bridges can connect to objects outside the host, but such a connection is not mandatory.

Report a bug

4.2.4. Bond

A *Bond* aggregates multiple NICs in a parallel manner into a single software device. Network bond devices can provide combined speed that is beyond single NIC speeds. Bonding also provides increased fault tolerance by increasing the number of failures required for networking to fail completely. The NICs that form a bond device must be of the same make and model in order to ensure that both devices support the same options and modes.

The packet dispersal algorithm for a bond is determined by the bonding mode used.

Bonding Modes

Red Hat Enterprise Virtualization uses mode 4 by default but supports the following common bonding modes:

- Mode 0 (round-robin policy) transmits packets to interfaces in sequential order. Packet transmission occurs in iterations that begin with the first available interface and end with the last available interface. Similarly, all subsequent iterations initiate with the first available interface. Mode 0 offers fault tolerance and load balancing to the network. However, mode 0 cannot be used in conjunction with bridges, therefore it is not compatible with virtual machine logical networks.
- Mode 1 (active-backup policy) sets all interfaces to the backup state while one remains active. Upon failure on the active interface, a backup interface replaces it as the only active interface in the bond. The MAC address of the bond in mode 1 is visible on only one port (the network adapter), to prevent confusion for the switch. Mode 1 provides fault tolerance and is supported in Red Hat Enterprise Virtualization.
- Mode 2 (XOR policy) selects an interface to transmit packages to based on the result of an XOR operation on the source and destination MAC addresses modulo NIC slave count. This calculation ensures that the same interface is selected for each destination MAC address used. Mode 2 provides fault tolerance and load balancing and is supported in Red Hat Enterprise Virtualization.
- Mode 3 (broadcast policy) transmits all packages to all interfaces. Mode 3 provides fault tolerance and is supported in Red Hat Enterprise Virtualization.

- Mode 4 (IEEE 802.3ad policy) creates aggregation groups for which included interfaces share the speed and duplex settings. Mode 4 uses all interfaces in the active aggregation group in accordance with the IEEE 802.3ad specification and is supported in Red Hat Enterprise Virtualization.
- Mode 5 (adaptive transmit load balancing policy) ensures the outgoing traffic distribution is according to the load on each interface and that the current interface receives all incoming traffic. If the interface assigned to receive traffic fails, another interface is assigned the receiving role instead. Mode 5 is supported in Red Hat Enterprise Virtualization.
- Mode 6 (adaptive load balancing policy)combines mode 5 (adaptive transmit load balancing policy) with receive load balancing for IPV4 traffic without any special switch requirements. ARP negotiation is used for balancing the receive load. Mode 6 cannot be used in conjunction with bridges, therefore it is not compatible with virtual machine logical networks.

Report a bug

4.2.5. Switch Configuration for Bonding

The following is an bond example configuration for a switch. Your switch configuration may look different.

```
interface Port-channel11
switchport access vlan 153
switchport mode access
spanning-tree portfast disable
spanning-tree bpduguard disable
spanning-tree guard root

interface GigabitEthernet0/16
switchport access vlan 153
switchport mode access
channel-group 11 mode active

interface GigabitEthernet0/17
switchport access vlan 153
switchport mode access
```

Important

For every type of switch it is important to set up the switch bonding with the *Link Aggregation Control Protocol* (LACP) protocol and *not* the Cisco *Port Aggregation Protocol* (PAgP) protocol.

Report a bug

4.2.6. Virtual Network Interface Controller (VNIC)

An NIC is the physical network interface controller for the host. A VNIC is a virtual NIC based on the physical NIC. Each host can have one or more NICs and each NIC can be a base for multiple VNICs. Every virtual machine with a network interface results in a new VNIC with a unique MAC address on the host where the virtual machine runs.

Each VNIC is then added to a network bridge device. The network bridge devices are how virtual machines are connected to virtual machine logical networks.

Running the **ifconfig** command on a Red Hat Enterprise Virtualization shows all of the VNICs that are associated with virtual machines on a host. Also visible are any network bridges that have been created to back logical networks, and any NICs used by the host.

```
[root@ecs-cloud-rhevh-01 ~]# ifconfig
eth0      Link encap:Ethernet  HWaddr E4:1F:13:B7:FD:D4
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:2527437 errors:0 dropped:0 overruns:0 frame:0
          TX packets:7353099 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1842636390 (1.7 GiB)  TX bytes:4527273914 (4.2 GiB)
          Interrupt:169 Memory:92000000-92012800

bond0     Link encap:Ethernet  HWaddr 00:1B:21:98:25:E4
          UP BROADCAST RUNNING MASTER MULTICAST  MTU:1500  Metric:1
          RX packets:1207008987 errors:0 dropped:2132 overruns:0 frame:0
          TX packets:1172475485 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:1564609462833 (1.4 TiB)  TX bytes:885715805671 (824.8
GiB)


rhevm     Link encap:Ethernet  HWaddr E4:1F:13:B7:FD:D4
          inet addr:10.64.14.122  Bcast:10.64.15.255  Mask:255.255.254.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:445040 errors:0 dropped:0 overruns:0 frame:0
          TX packets:4721866 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:41575335 (39.6 MiB)  TX bytes:4171361904 (3.8 GiB)

vnet000 Link encap:Ethernet  HWaddr FE:1A:4A:40:0E:04
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:477233 errors:0 dropped:0 overruns:0 frame:0
          TX packets:630027 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:500
          RX bytes:123257049 (117.5 MiB)  TX bytes:387924090 (369.9 MiB)

vnet001 Link encap:Ethernet  HWaddr FE:1A:4A:40:0E:30
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1642 errors:0 dropped:0 overruns:0 frame:0
          TX packets:120753 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:500
          RX bytes:318222 (310.7 KiB)  TX bytes:14323345 (13.6 MiB)

vnet002 Link encap:Ethernet  HWaddr FE:1A:4A:40:0E:2E
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:239673 errors:0 dropped:0 overruns:0 frame:0
          TX packets:555398 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:500
          RX bytes:17514233 (16.7 MiB)  TX bytes:620136453 (591.4 MiB)
```

The console output from the **ifconfig** shows one bond device, **bond0**; one Ethernet NIC **eth0**; one network bridge: **rhevm**; and a number of VNICs that are associated virtual machine network interfaces using **virtio drivers**.

The VNICs displayed in the given console output are all members of a network bridge device and logical network. Bridge membership can be displayed using the **brctl show** command:

```
[root@ecs-cloud-rhevh-01 ~]# brctl show
bridge name bridge id  STP enabled interfaces
rhevm  8000.e41f13b7fdd4 no  vnet002
       vnet001
       vnet000
       eth0
```

The console output from the **brctl show** shows that the virtio VNICs are members of the **rhevm** bridge.All of the virtual machines that the VNICs are associated with are connected to the **rhevm** logical network network. The **eth0** NIC is also a member of the **rhevm** bridge. The **eth0** device is cabled to a switch that provides connectivity beyond the host.

Report a bug

4.2.7. Virtual LAN (VLAN)

A *VLAN (Virtual LAN)* is an attribute that can be applied to network packets. Network packets can be "tagged" into a numbered VLAN. A VLAN is a security feature used to completely isolate network traffic at the switch level. VLANs are completely separate and mutually exclusive. The Red Hat Enterprise Virtualization Manager is VLAN aware and able to tag and redirect VLAN traffic, however VLAN implementation requires a switch that supports VLANs.

At the switch level, ports are assigned a VLAN designation. A switch applies a VLAN tag to traffic originating from a particular port, marking the traffic as part of a VLAN, and ensures that responses carry the same VLAN tag. A VLAN can extend across multiple switches. VLAN tagged network traffic on a switch is completely undetectable except by machines connected to a port designated with the correct VLAN. A given port can be tagged into multiple VLANs, which allows traffic from multiple VLANs to be sent to a single port, to be deciphered using software on the machine that receives the traffic.

Report a bug

4.3. Networking in data centers and clusters

4.3.1. Cluster Networking

Cluster level networking objects include:

- Clusters
- Logical Networks

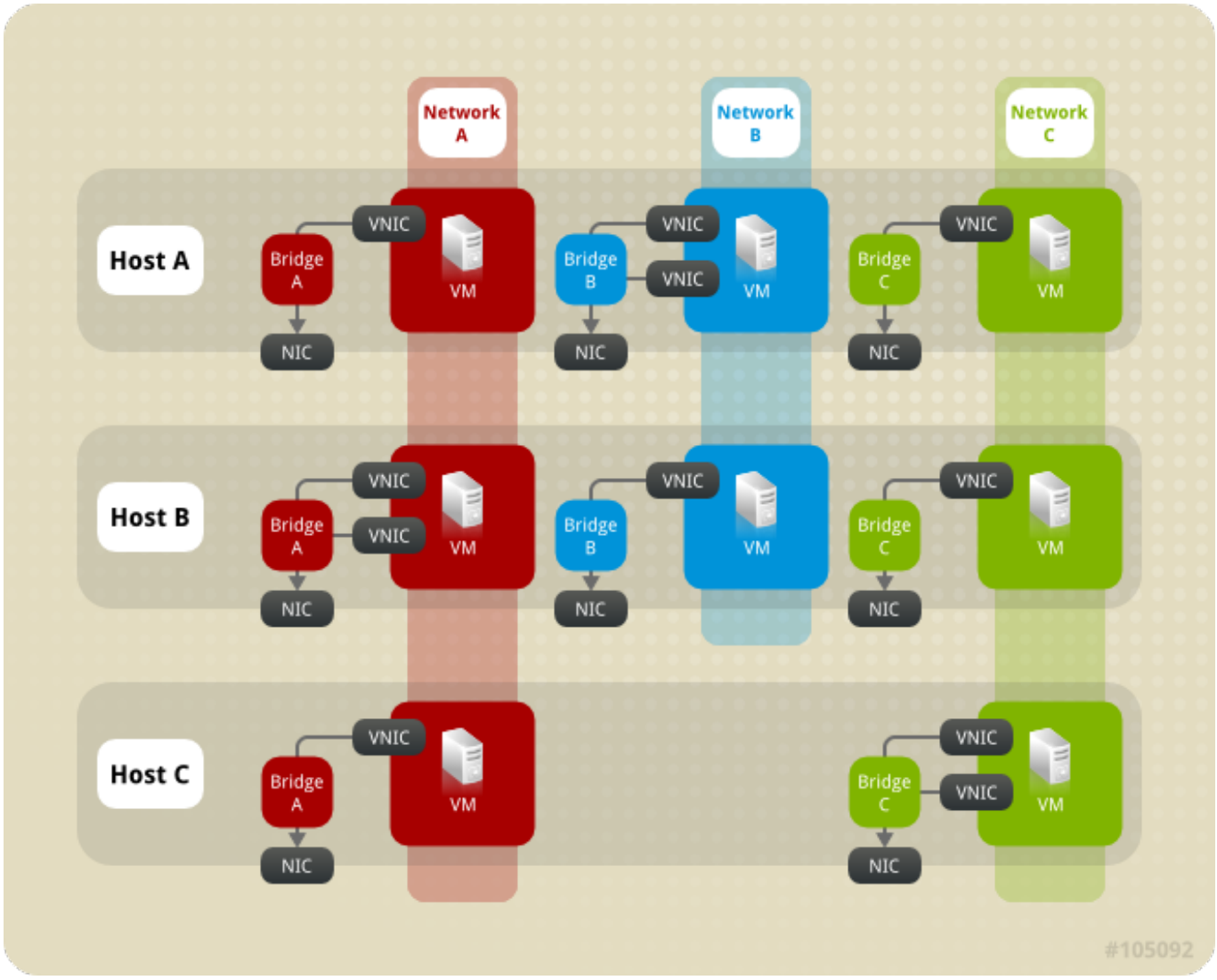


Figure 4.1. Networking within a cluster

A data center is a logical grouping of multiple clusters and each cluster is a logical group of multiple hosts. Figure 4.1, “Networking within a cluster” depicts the contents of a single cluster.

Hosts in a cluster all have access to the same storage domains. Hosts in a cluster also have logical networks applied at the cluster level. For a virtual machine logical network to become operational for use with virtual machines, the network must be defined and implemented for each host in the cluster using the Red Hat Enterprise Virtualization Manager. Other logical network types can be implemented on only the hosts that use them.

Report a bug

4.3.2. Logical Networks

Logical networking allows the Red Hat Enterprise Virtualization environment to separate network traffic by type. For example, the **rhevm** network is created by default during the installation of the Red Hat Enterprise Virtualization to be used for management communication between the Manager and hosts. A typical use for logical networks is to group network traffic with similar requirements and usage together. In many cases, a storage network and a display network are created by an administrator to isolate traffic of each respective type for optimization and troubleshooting.

The types of logical network are:

- logical networks that carry virtual machine network traffic,

- logical networks that do not carry virtual machine network traffic,
- optional logical networks,
- and required networks.

All logical networks can either be required or optional.

Logical networks are defined at the data center level, and added to a host. For a required logical network to be operational, it must be implemented for every host in a given cluster.

Each virtual machine logical network in a Red Hat Enterprise Virtualization environment is backed by a network bridge device on a host. So when a new virtual machine logical network is defined for a cluster, a matching bridge device must be created on each host in the cluster before the logical network can become operational to be used by virtual machines. Red Hat Enterprise Virtualization Manager automatically creates required bridges for virtual machine logical networks.

The bridge device created by the Red Hat Enterprise Virtualization Manager to back a virtual machine logical network is associated with a host network interface. If the host network interface that is part of a bridge has network connectivity, then any network interfaces that are subsequently included in the bridge share the network connectivity of the bridge. When virtual machines are created and placed on a particular logical network, their virtual network cards are included in the bridge for that logical network. Those virtual machines can then communicate with each other and with other objects that are connected to the bridge.

Logical networks not used for virtual machine network traffic are associated with host network interfaces directly.

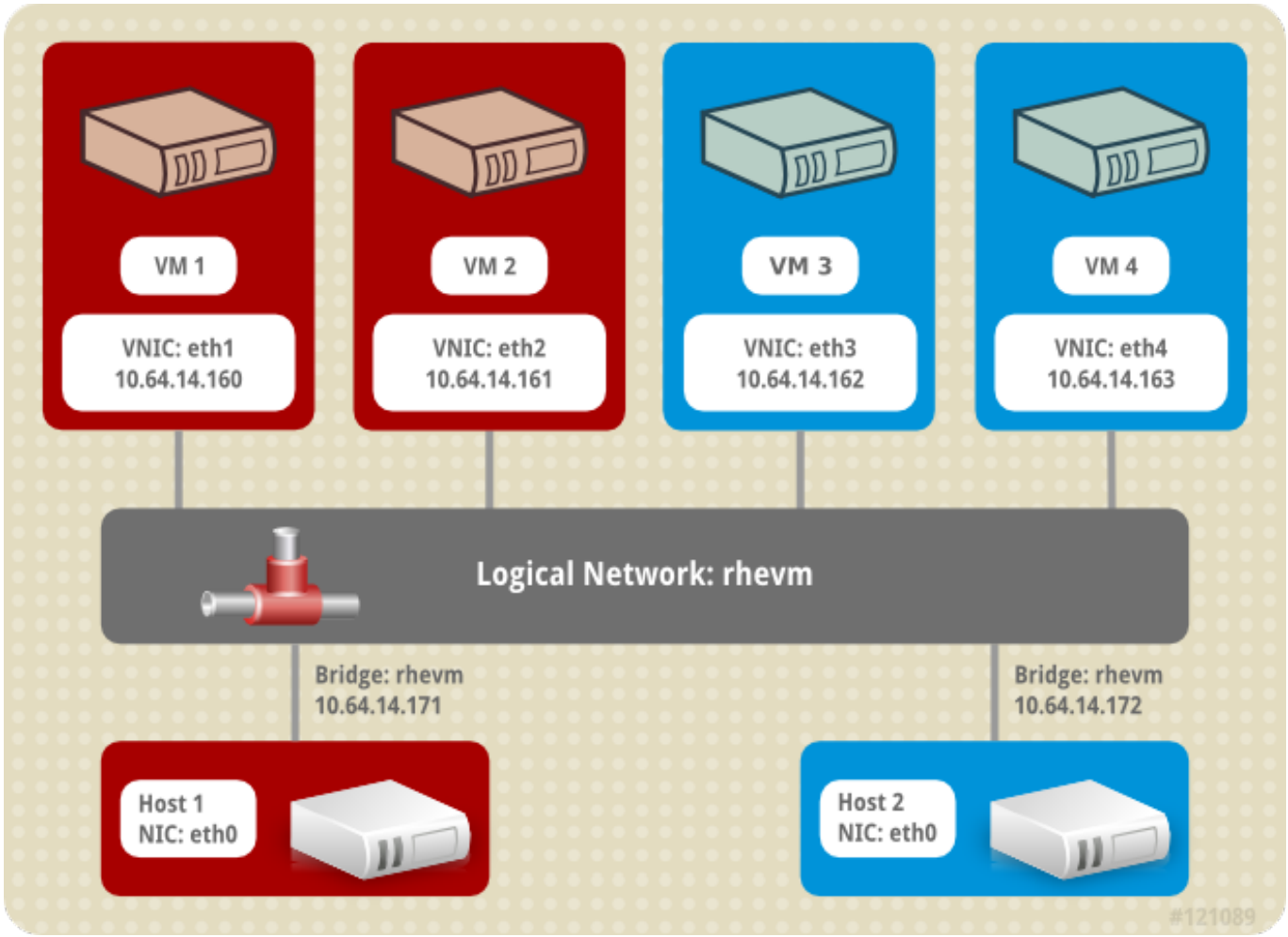


Figure 4.2. The rhevm logical network.

Example 4.1. Example usage of a logical network.

There are two hosts called Red and White in a cluster called Pink in a data center called Purple. Both Red and White have been using the default logical network, **rhevm** for all networking functions. The system administrator responsible for Pink decides to isolate network testing for a web server by placing the web server and some client virtual machines on a separate logical network. She decides to call the new logical network **network_testing**.

First, she defines the logical network for the Purple data center. She then applies it to the Pink cluster. Logical networks must be implemented on a host in maintenance mode. So, the administrator first migrates all running virtual machines to Red, and puts White in maintenance mode. Then she edits the **Network** associated with the physical network interface that will be included in the bridge. The **Link Status** for the selected network interface will change from **Down** to **Non-Operational**. The non-operational status is because the corresponding bridge must be setup in all hosts in the cluster by adding a physical network interface on each host in the Pink cluster to the **network_testing** network. Next she activates White, migrates all of the running virtual machines off of Red, and repeats the process for Red.

When both White and Red both have the **network_testing** logical network bridged to a physical network interface, the **network_testing** logical network becomes **Operational** and is ready to be used by virtual machines.

Report a bug

4.3.3. Required Networks, Optional Networks, and Virtual Machine Networks

Red Hat Enterprise Virtualization 3.1 and higher distinguishes between required networks and optional networks.

Required networks must be applied to all hosts in a cluster for the cluster and network to be **Operational**. Logical networks are added to clusters as **Required** networks by default.

When a required network becomes non-operational, the virtual machines running on the network are fenced and migrated to another host. This is beneficial if you have machines running mission critical workloads.

When a non-required network becomes non-operational, the virtual machines running on the network are not migrated to another host. This prevents unnecessary I/O overload caused by mass migrations.

Optional networks are those logical networks that have not been explicitly declared **Required** networks. Optional networks can be implemented on only the hosts that use them. The presence or absence of these networks does not affect the **Operational** status of a host.

Use the **Manage Networks** button to change a network's **Required** designation.

Virtual machine networks (called a **VM network** in the user interface) are logical networks designated to carry only virtual machine network traffic. Virtual machine networks can be required or optional.

Note

A virtual machine with a network interface on an optional virtual machine network will not start on a host without the network.

Report a bug

4.4. Networking in hosts and virtual machines

4.4.1. Virtual Machine Connectivity

In Red Hat Enterprise Virtualization, a virtual machine has its NIC put on a logical network at the time that the virtual machine is created. From that point, the virtual machine is able to communicate with any other destination on the same network.

From the host perspective, when a virtual machine is put on a logical network, the VNIC that backs the virtual machine's NIC is added as a member to the bridge device for the logical network. For example, if a virtual machine is on the **rhevm** logical network, its VNIC is added as a member of the **rhevm** bridge of the host on which that virtual machine runs.

Report a bug

4.4.2. Port Mirroring

Port mirroring copies layer 3 network traffic on given a logical network and host to a virtual interface on a virtual machine. This virtual machine can be used for network debugging and tuning, intrusion detection, and monitoring the behavior of other virtual machines on the same host and logical network.

The only traffic copied is internal to one logical network on one host. There is no increase on traffic on the network external to the host; however a virtual machine with port mirroring enabled uses more host CPU and RAM than other virtual machines.

Enable and disable port mirroring by editing network interfaces on virtual machines.

Port mirroring requires an IPv4 IP address.

Important

You should be aware that enabling port mirroring reduces the privacy of any other network users.

Report a bug

4.4.3. Host Networking Configurations

Common types of networking configurations for Red Hat Enterprise Virtualization hosts include:

- Bridge and NIC configuration.
- Bridge, VLAN, and NIC configuration.

- Bridge, Bond, and VLAN configuration.
- Multiple Bridge, Multiple VLAN, and NIC configuration.

Report a bug

4.4.4. Bridge Configuration

The simplest host configuration in Red Hat Enterprise Virtualization is the Bridge and NIC configuration. As Figure 4.3, “Bridge and NIC configuration” depicts, this configuration uses a bridge to connect one or more virtual machines (or guests) to the host's NIC.

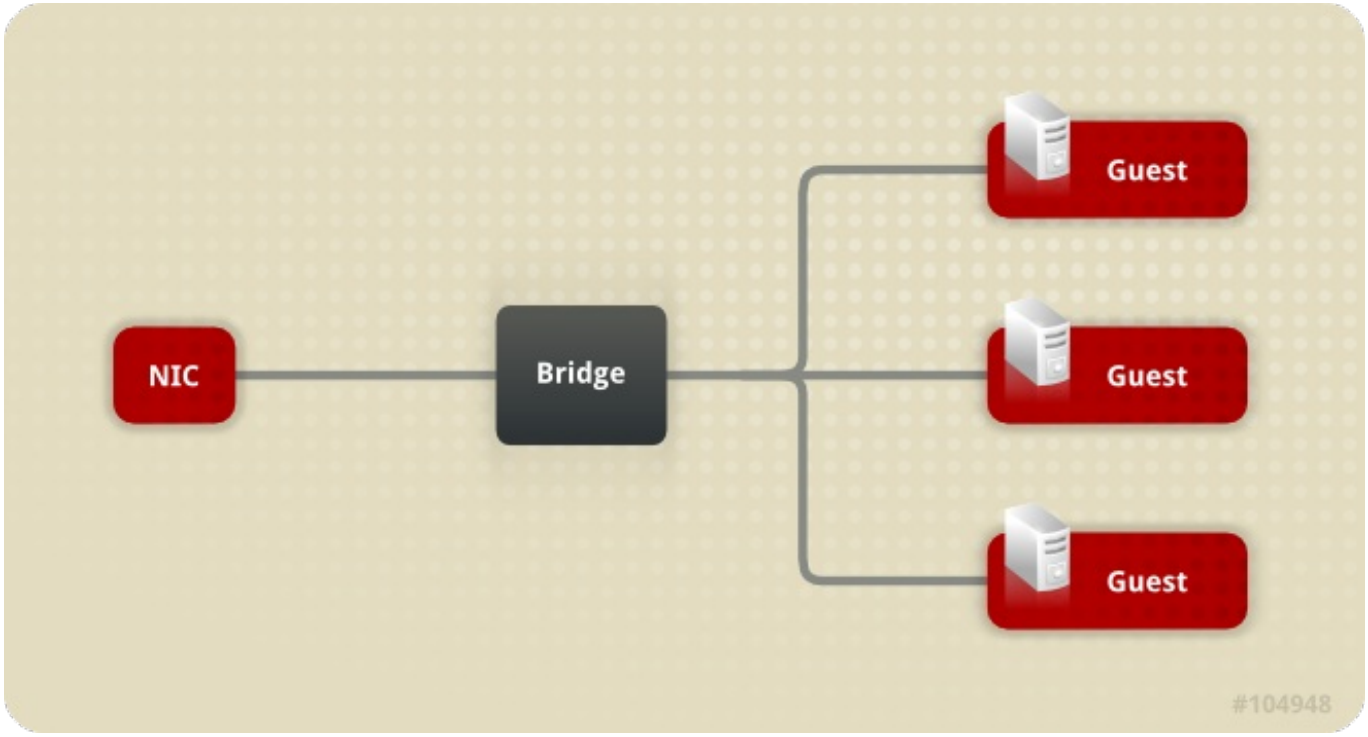


Figure 4.3. Bridge and NIC configuration

An example of this configuration is the automatic creation of the bridge **rhevm** when the Red Hat Enterprise Virtualization Manager installs. On installation, the Red Hat Enterprise Virtualization Manager installs **VDSM** on the host. The **VDSM** installation process creates the bridge **rhevm**. The **rhevm** bridge then obtains the **IP** address of the host to enable management communication for the host.

Report a bug

4.4.5. VLAN Configuration

Figure 4.4, “Bridge, VLAN, and NIC configuration” depicts an alternative configuration that includes a virtual LAN (VLAN) to connect the host NIC and bridge.

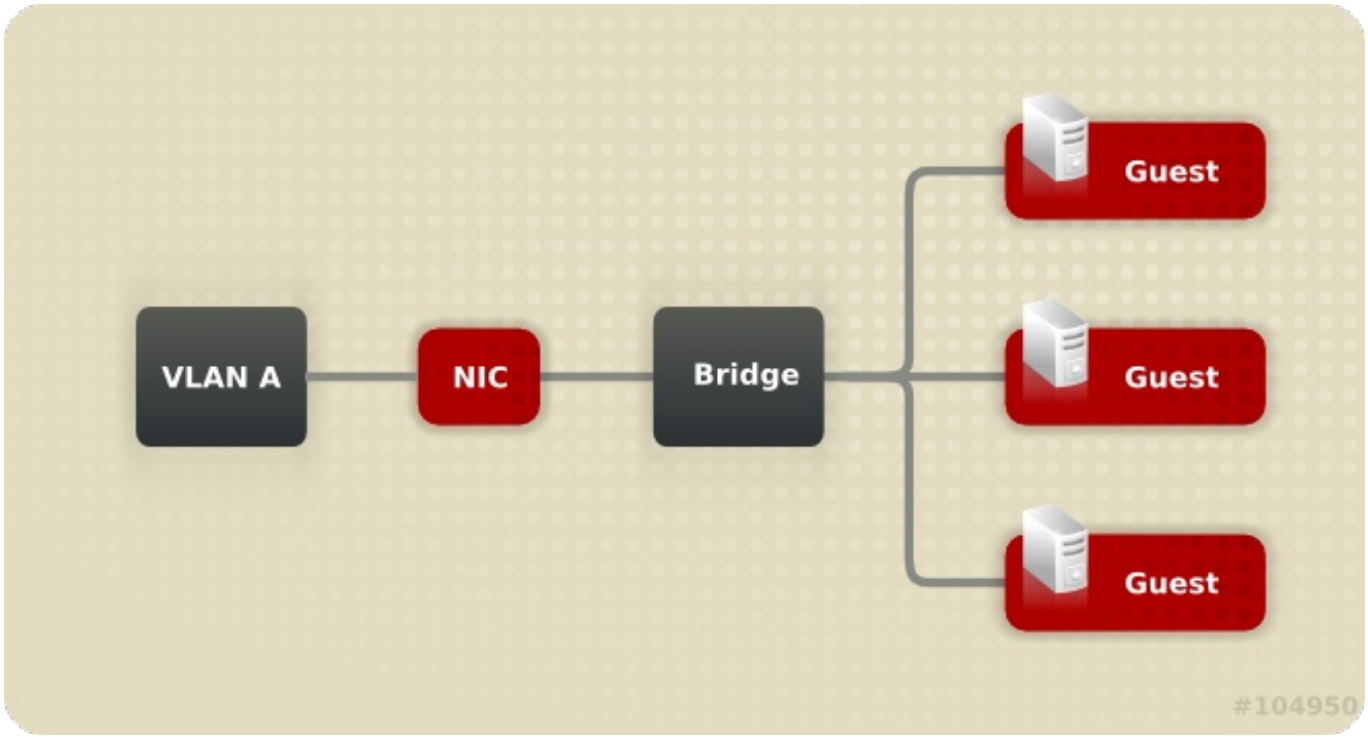


Figure 4.4. Bridge, VLAN, and NIC configuration

A VLAN is included to provide a secure channel for data transfer over this network and also to support the option to connect multiple bridges to a single NIC using multiple VLANs.

Report a bug

4.4.6. Bridge and Bond Configuration

Figure 4.5, “Bridge, Bond, and NIC configuration” displays a configuration that includes a bond to connect multiple host NICs to the same bridge and network.

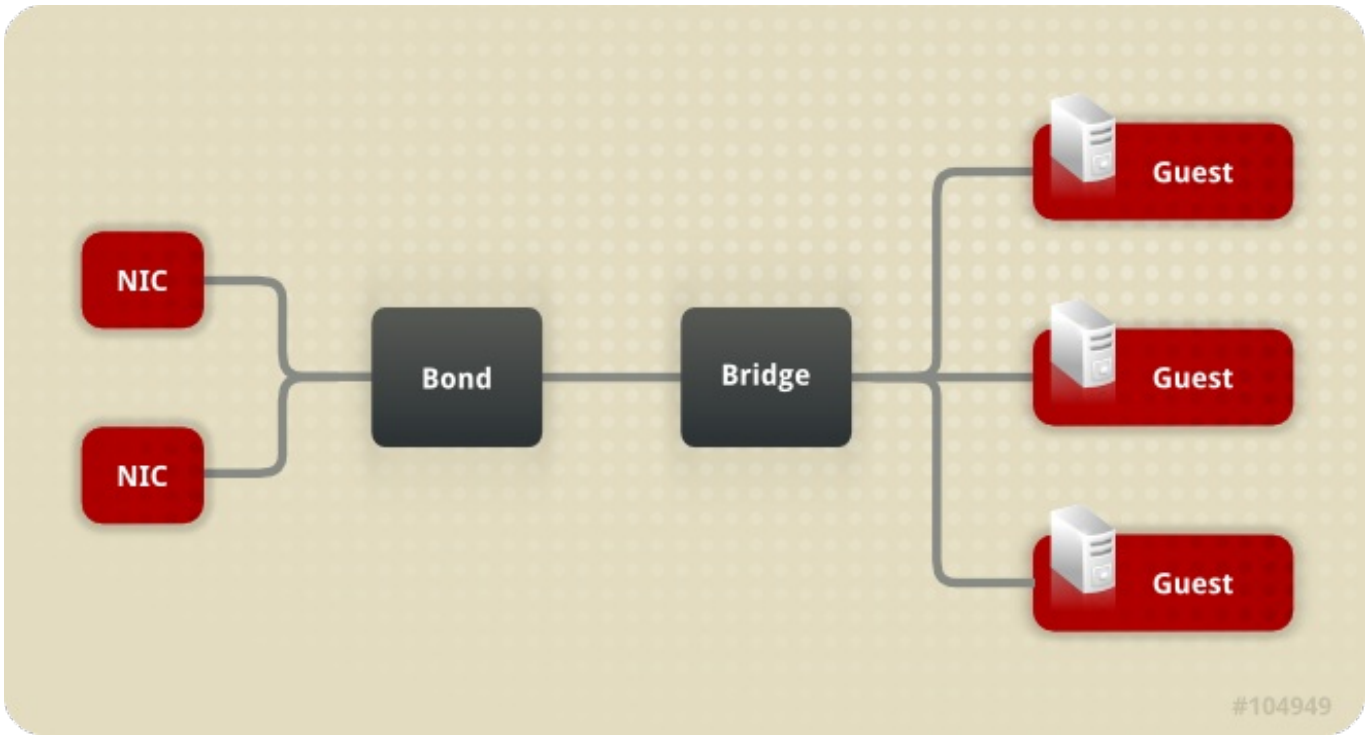


Figure 4.5. Bridge, Bond, and NIC configuration

The included bond creates a logical link that combines the two (or more) physical Ethernet links. The resultant benefits include NIC fault tolerance and potential bandwidth extension, depending on the bonding mode.

Report a bug

4.4.7. Multiple Bridge, Multiple VLAN, and NIC Configuration

Figure 4.6, “Multiple Bridge, Multiple VLAN, and NIC configuration” depicts a configuration that connects a single NIC to two VLANs. This presumes that the network switch has been configured to pass network traffic that has been tagged into one of the two VLANs to one NIC on the host. The host uses two VNICs to separate VLAN traffic, one for each VLAN. Traffic tagged into either VLAN then connects to a separate bridge by having the appropriate VNIC as a bridge member. Each bridge is in turn connected to by multiple virtual machines.

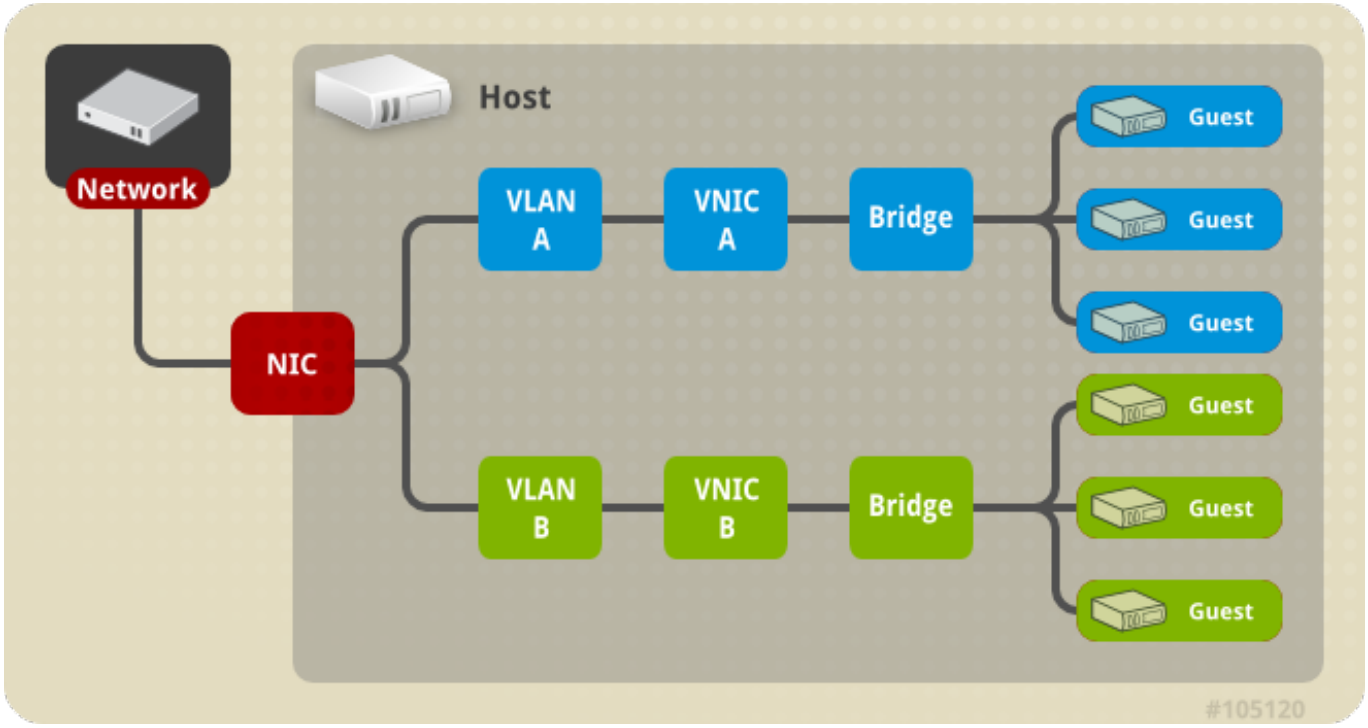


Figure 4.6. Multiple Bridge, Multiple VLAN, and NIC configuration

Report a bug

4.4.8. Multiple Bridge, Multiple VLAN, and Bond Configuration

Figure 4.7, “Multiple Bridge, Multiple VLAN, and Multiple NIC with Bond connection” displays a configuration that bonds multiple NICs to facilitate a connection with multiple VLANs.

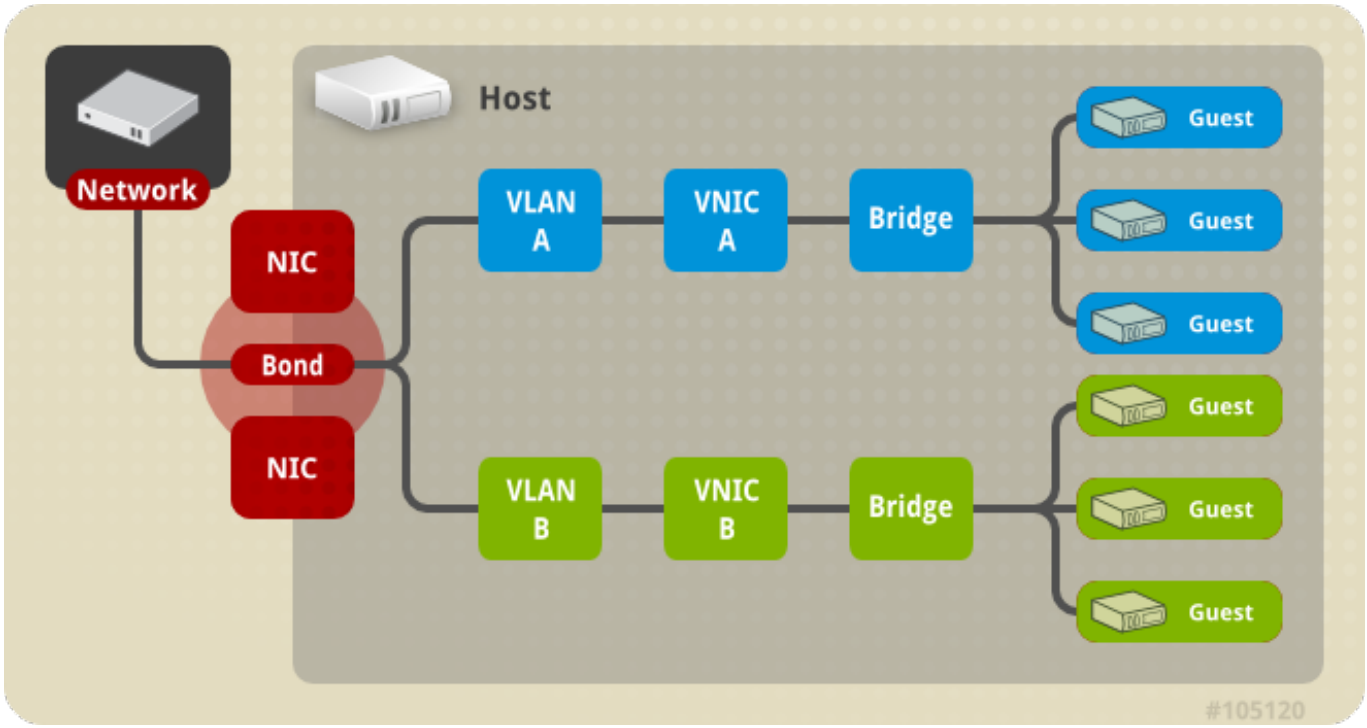


Figure 4.7. Multiple Bridge, Multiple VLAN, and Multiple NIC with Bond connection

Each VLAN in this configuration is defined over the bond connecting the NICs. Each VLAN connects to an individual bridge and each bridge connects to one or more guests.

Report a bug

Chapter 5. Power management and fencing

5.1. Introduction to Power Management and Fencing

The Red Hat Enterprise Virtualization environment is most flexible and resilient when power management and fencing have been configured. Power management allows the Red Hat Enterprise Virtualization Manager to control host power cycle operations, most importantly to reboot hosts on which problems have been detected. Fencing is used to isolate problem hosts from a functional Red Hat Enterprise Virtualization environment by rebooting them, in order to prevent performance degradation. Fenced hosts can then be returned to responsive status through administrator action and be reintegrated into the environment.

Power management and fencing make use of special dedicated hardware in order to restart hosts independently of host operating systems. The Red Hat Enterprise Virtualization Manager connects to a power management devices using a network IP address or hostname. In the context of Red Hat Enterprise Virtualization, a power management device and a fencing device are the same thing.

Report a bug

5.2. Power Management by Proxy in Red Hat Enterprise Virtualization

The Red Hat Enterprise Virtualization Manager does not communicate directly with fence agents. Instead, the Manager uses a proxy to send power management commands to a host power management device. The Manager uses VDSM to execute power management device actions, so another host in the environment is used as a fencing proxy.

You can select between:

- Any host in the same cluster as the host requiring fencing.
- Any host in the same data center as the host requiring fencing.

A viable fencing proxy host has a status of either *UP* or *Maintenance*.

Report a bug

5.3. Power Management

The Red Hat Enterprise Virtualization Manager is capable of rebooting hosts that have entered a non-operational or non-responsive state, as well as preparing to power off under-utilized hosts to save power. This functionality depends on a properly configured power management device. The Red Hat Enterprise Virtualization environment supports the following power management devices:

- *American Power Conversion* (apc).
- Bladecenter.
- *Cisco Unified Computing System* (cisco_ucs).
- *Dell Remote Access Card 5* (drac5).
- *Electronic Power Switch* (eps).
- *Integrated Lights Out* (ilo, ilo2, ilo3, ilo4).
- *Intelligent Platform Management Interface* (ipmilan).

- *Remote Supervisor Adapter* (rsa).
- rsb.
- *Western Telematic, Inc* (wti).

Note

APC 5.x power management devices are not supported by the **apc** fence agent. Use the **apc_snmp** fence agent instead.

In order to communicate with the listed power management devices, the Red Hat Enterprise Virtualization Manager makes use of *fence agents*. The Red Hat Enterprise Virtualization Manager allows administrators to configure a fence agent for the power management device in their environment with parameters the device will accept and respond to. Basic configuration options can be configured using the graphical user interface. Special configuration options can also be entered, and are passed un-parsed to the fence device. Special configuration options are specific to a given fence device, while basic configuration options are for functionalities provided by all supported power management devices. The basic functionalities provided by all power management devices are:

- **Status**: check the status of the host.
- **Start**: power on the host.
- **Stop**: power down the host.
- **Restart**: restart the host. Actually implemented as stop, wait, status, start, wait, status.

Best practice is to test the power management configuration once when initially configuring it, and occasionally after that to ensure continued functionality.

Resilience is provided by properly configured power management devices in all of the hosts in an environment. Fencing agents allow the Red Hat Enterprise Virtualization Manager to communicate with host power management devices to bypass the operating system on a problem host, and isolate the host from the rest of its environment by rebooting it. The Manager can then reassign the SPM role, if it was held by the problem host, and safely restart any highly available virtual machines on other hosts.

Report a bug

5.4. Fencing

In the context of the Red Hat Enterprise Virtualization environment, fencing is a host reboot initiated by the Manager using a fence agent and performed by a power management device. Fencing allows a cluster to react to unexpected host failures as well as enforce power saving, load balancing, and virtual machine availability policies.

Fencing ensures that the role of Storage Pool Manager (SPM) is always assigned to a functional host. If the a fenced host was the SPM, the SPM role is relinquished and reassigned to a responsive host. Because the host with the SPM role is the only host that is able to write data domain structure metadata, a non-responsive, un-fenced SPM host causes its environment to lose the ability to create and destroy virtual disks, take snapshots, extend logical volumes, and all other actions that require changes to data domain structure metadata.

When a host becomes non-responsive, all of the virtual machines that are currently running on that

host can also become non-responsive. However, the non-responsive host retains the lock on the virtual machine hard disk images for virtual machines it is running. Attempting to start a virtual machine on a second host and assign the second host write privileges for the virtual machine hard disk image can cause data corruption.

Fencing allows the Red Hat Enterprise Virtualization Manager to assume that the lock on a virtual machine hard disk image has been released; the Manager can use a fence agent to confirm that the problem host has been rebooted. When this confirmation is received, the Red Hat Enterprise Virtualization Manager can start a virtual machine from the problem host on another host without risking data corruption. Fencing is the basis for highly-available virtual machines. A virtual machine that has been marked highly-available can not be safely started on an alternate host without the certainty that doing so will not cause data corruption.

When a host becomes non-responsive, the Red Hat Enterprise Virtualization Manager allows a grace period of thirty (30) seconds to pass before any action is taken, to allow the host to recover from any temporary errors. If the host has not become responsive by the time the grace period has passed, the Manager automatically begins to mitigate any negative impact from the non-responsive host. The Manager uses the fencing agent for the power management card on the host to stop the host, confirm it has stopped, start the host, and confirm that the host has been started. When the host finishes booting, it attempts to rejoin the cluster that it was a part of before it was fenced. If the issue that caused the host to become non-responsive has been resolved by the reboot, then the host is automatically set to **Up** status and is once again capable of starting and hosting virtual machines.

Report a bug

5.5. Soft-Fencing Hosts

Sometimes a host becomes non-responsive due to an unexpected problem, and though VDSM is unable to respond to requests, the virtual machines that depend upon VDSM remain alive and accessible. In these situations, restarting VDSM returns VDSM to a responsive state and resolves this issue.

Red Hat Enterprise Virtualization 3.3 introduces "soft-fencing over SSH". Prior to Red Hat Enterprise Virtualization 3.3, non-responsive hosts were fenced only by external fencing devices. In Red Hat Enterprise Virtualization 3.3, the fencing process has been expanded to include "SSH Soft Fencing", a process whereby the Manager attempts to restart VDSM via SSH on non-responsive hosts. If the Manager fails to restart VDSM via SSH, the responsibility for fencing falls to the external fencing agent if an external fencing agent has been configured.

Soft-fencing over SSH works as follows. Fencing must be configured and enabled on the host, and a valid proxy host (a second host, in an UP state, in the data center) must exist. When the connection between the Manager and the host times out, the following happens:

1. On the first network failure, the status of the host changes to "connecting".
2. The Manager then makes three attempts to ask VDSM for its status, or it waits for an interval determined by the load on the host. The formula for determining the length of the interval is configured by the configuration values TimeoutToResetVdsInSeconds (the default is 60 seconds) + [DelayResetPerVmInSeconds (the default is 0.5 seconds)]*(the count of running vms on host) + [DelayResetForSpmInSeconds (the default is 20 seconds)] * 1 (if host runs as SPM) or 0 (if the host does not run as SPM). To give VDSM the maximum amount of time to respond, the Manager chooses the longer of the two options mentioned above (three attempts to retrieve the status of VDSM or the interval determined by the above formula).
3. If the host does not respond when that interval has elapsed, **`vdsm restart`** is executed via SSH.

4. If **`vdsm restart`** does not succeed in re-establishing the connection between the host and the Manager, the status of the host changes to **`Non Responsive`** and, if power management is configured, fencing is handed off to the external fencing agent.

Note

Soft-fencing over SSH can be executed on hosts that have no power management configured. This is distinct from "fencing": fencing can be executed only on hosts that have power management configured.

Report a bug

5.6. Using Multiple Power Management Fencing Agents

Prior to Red Hat Enterprise Virtualization 3.2, a host with power management configured only recognized one fencing agent. Fencing agents configured on version 3.1 and earlier, and single agents, are treated as primary agents. The secondary agent is valid when there are two fencing agents, for example for dual-power hosts in which each power switch has two agents connected to the same power switch. Agents can be of the same or different types.

Having multiple fencing agents on a host increases the reliability of the fencing procedure. For example, when the sole fencing agent on a host fails, the host will remain in a non-operational state until it is manually rebooted. The virtual machines previously running on the host will be suspended, and only fail over to another host in the cluster after the original host is manually fenced. With multiple agents, if the first agent fails, the next agent can be called.

When two fencing agents are defined on a host, they can be configured to use a *concurrent* or *sequential* flow:

- **Concurrent**: Both primary and secondary agents have to respond to the Stop command for the host to be stopped. If one agent responds to the Start command, the host will go up.
- **Sequential**: To stop or start a host, the primary agent is used first, and if it fails, the secondary agent is used.

Report a bug

Chapter 6. Load Balancing, scheduling, and migration

6.1. Load Balancing, Scheduling, and Migration

Individual hosts have finite hardware resources, and are susceptible to failure. To mitigate against failure and resource exhaustion, hosts are grouped into clusters, which are essentially a grouping of shared resources. A Red Hat Enterprise Virtualization environment responds to changes in demand for host resources using load balancing policy, scheduling, and migration. The Manager is able to ensure that no single host in a cluster is responsible for all of the virtual machines in that cluster. Conversely, the Manager is able to recognize an underutilized host, and migrate all virtual machines off of it, allowing an administrator to shut down that host to save power.

Available resources are checked as a result of three events:

- Virtual machine start - Resources are checked to determine on which host a virtual machine will start.
- Virtual machine migration - Resources are checked in order to determine an appropriate target host.
- Time elapses - Resources are checked at a regular interval to determine whether individual host load is in compliance with cluster load balancing policy.

The Manager responds to changes in available resources by using the load balancing policy for a cluster to schedule the migration of virtual machines from one host in a cluster to another. The relationship between load balancing policy, scheduling, and virtual machine migration are discussed in the following sections.

Report a bug

6.2. Load Balancing Policy

Load balancing policy is set for a cluster, which includes one or more hosts that may each have different hardware parameters and available memory. The Red Hat Enterprise Virtualization Manager uses a load balancing policy to determine which host in a cluster to start a virtual machine on. Load balancing policy also allows the Manager determine when to move virtual machines from over-utilized hosts to under-utilized hosts.

The load balancing process runs once every minute for each cluster in a data center. It determines which hosts are over-utilized, which are hosts under-utilized, and which are valid targets for virtual machine migration. The determination is made based on the load balancing policy set by an administrator for a given cluster. There are three load balancing policies: **`None`**, **`Even Distribution`**, and **`Power Saving`**.

Report a bug

6.3. Load Balancing Policy: None

If no load balancing policy is selected, virtual machines are started on the host within a cluster with the lowest CPU utilization and available memory. To determine CPU utilization a combined metric is used that takes into account the virtual CPU count and the CPU usage percent. This approach is the least dynamic, as the only host selection point is when a new virtual machine is started. Virtual machines are not automatically migrated to reflect increased demand on a host.

An administrator must decide which host is an appropriate migration target for a given virtual machine. Virtual machines can also be associated with a particular host using *pinning*. Pinning prevents a virtual machine from being automatically migrated to other hosts. For environments where resources are highly consumed, manual migration is the best approach.

Report a bug

6.4. Load Balancing Policy: Even Distribution

An even distribution load balancing policy selects the host for a new virtual machine according to lowest CPU utilization. The maximum service level is the maximum CPU utilization that is allowed for hosts in a cluster, beyond which environment performance will degrade. The even distribution policy allows an administrator to set a maximum service level for running virtual machines. The length of time a host is allowed to continue at this maximum service level before the Red Hat Enterprise Virtualization Manager intervenes is also set by an administrator. If a host has reached the maximum service level and stays there for more then the set time, virtual machines on that host are migrated one by one to the host in the cluster that has the lowest CPU utilization. Host resources are checked once per minute, and one virtual machine is migrated at a time until the host CPU utilization is below the maximum service threshold.

Report a bug

6.5. Load Balancing Policy: Power Saving

A power saving load balancing policy selects the host for a new virtual machine according to lowest CPU utilization. The maximum service level is the maximum CPU utilization that is allowed for hosts in a cluster, beyond which environment performance will degrade. The minimum service level is the minimum CPU utilization allowed before the continued operation of a host is considered an inefficient use of electricity. The even distribution policy allows an administrator to set a maximum and minimum service level for running virtual machines. The length of time a host is allowed to continue at this maximum or minimum service level before the Red Hat Enterprise Virtualization Manager intervenes is also set by an administrator. If a host has reached the maximum service level and stays there for more than the set time, the virtual machines on that host are migrated one by one to the host that has the lowest CPU utilization. The process continues until the host CPU utilization is below maximum service level. If a host CPU utilization falls below the minimum service level the virtual machines are migrated to other hosts in the cluster if their maximum service level permits. When an under-utilized host is cleared of its remaining virtual machines, it can be shut down by an administrator to preserve power.

Report a bug

6.6. Scheduling

In Red Hat Enterprise Virtualization, scheduling refers to the way the Red Hat Enterprise Virtualization Manager selects a host in a cluster as the target for a new or migrated virtual machine.

For a host to be eligible to start a virtual machine or accept a migrated virtual machine from another host, it must have enough free memory and CPUs to support the requirements of the virtual machine being started on or migrated to it. If multiple hosts are eligible targets, one will be selected based on the load balancing policy for the cluster. For example, if an even distribution policy is in effect, the Manager chooses the host with the lowest CPU utilization. If the power saving policy is in effect, the host with the lowest CPU utilization between the maximum and minimum service levels will be

selected. The Storage Pool Manager (SPM) status of a given host also affects eligibility as a target for starting virtual machines or virtual machine migration. A non-SPM host is a preferred target host, for instance, the first virtual machine started in a cluster will not run on the SPM host if the SPM role is held by a host in that cluster.

See Also:

- Section 3.1.8, “The Storage Pool Manager”

Report a bug

6.7. Migration

The Red Hat Enterprise Virtualization Manager uses migration to enforce load balancing policies for a cluster. Virtual machine migration takes place according to the load balancing policy for a cluster and current demands on hosts within a cluster. Migration can also be configured to automatically occur when a host is fenced or moved to maintenance mode. The Red Hat Enterprise Virtualization Manager first migrates virtual machines with the lowest CPU utilization. This is calculated as a percentage, and does not take into account RAM usage or I/O operations, except as I/O operations affect CPU utilization. If there are more than one virtual machines with the same CPU usage, the one that will be migrated first is the first virtual machine returned by the database query run by the Red Hat Enterprise Virtualization Manager to determine virtual machine CPU usage.

Migration Statistics

A bandwidth limit of 30 Mbps is imposed on each virtual machine migration. A migration will timeout after a certain amount of time has passed. The timeout happens after either 300 seconds, or after the amount of virtual machine memory in Mb divided by 2048 Mb multiplied by 300 seconds, whichever is larger.

By default, concurrent outgoing migrations are limited to one per CPU core per host, or 5, whichever is smaller.

Report a bug

Chapter 7. Directory services

7.1. Directory Services

The Red Hat Enterprise Virtualization platform relies on directory services for user authentication and authorization. Interactions with all Manager interfaces, including the User Portal, Power User Portal, Administration Portal, and REST API are limited to authenticated, authorized users. Virtual machines within the Red Hat Enterprise Virtualization environment can use the same directory services to provide authentication and authorization, however they must be configured to do so. The currently supported providers of directory services for use with the Red Hat Enterprise Virtualization Manager are *Identity Management* (IdM), *Red Hat Directory Server 9* (RHDS), *Active Directory* (AD), and *OpenLDAP*. The Red Hat Enterprise Virtualization Manager interfaces with the directory server for:

- Portal logins (User, Power User, Administrator, REST API).
- Queries to display user information.
- Adding the Manager to a domain.

Authentication is the verification and identification of a party who generated some data, and of the integrity of the generated data. A principal is the party whose identity is verified. The verifier is the party who demands assurance of the principal's identity. In the case of Red Hat Enterprise Virtualization, the Manager is the verifier and a user is a principal. Data integrity is the assurance that the data received is the same as the data generated by the principal.

Confidentiality and authorization are closely related to authentication. Confidentiality protects data from disclosure to those not intended to receive it. Strong authentication methods can optionally provide confidentiality. Authorization determines whether a principal is allowed to perform an operation. Red Hat Enterprise Virtualization uses directory services to associate users with roles and provide authorization accordingly. Authorization is usually performed after the principal has been authenticated, and may be based on information local or remote to the verifier.

During installation, a local, internal domain is automatically configured for administration of the Red Hat Enterprise Virtualization environment. After the installation is complete, more domains can be added.

Report a bug

7.2. Local Authentication: Internal Domain

The Red Hat Enterprise Virtualization Manager creates a limited, internal administration domain during installation. This domain is not the same as an AD or IdM domain, because it exists based on a key in the Red Hat Enterprise Virtualization PostgreSQL database rather than as a directory service user on a directory server. The internal domain is also different from an external domain because the internal domain will only have one user: the **admin@internal** user. Taking this approach to initial authentication allows Red Hat Enterprise Virtualization to be evaluated without requiring a complete, functional directory server, and ensures an administrative account is available to troubleshoot any issues with external directory services.

The admin@internal user is for the initial configuration of an environment. This includes installing and accepting hosts, adding external AD or IdM authentication domains, and delegating permissions to users from external domains.

Report a bug

7.3. Remote Authentication Using GSSAPI

7.3. Remote Authentication Using GSSAPI

In the context of Red Hat Enterprise Virtualization, remote authentication refers to authentication that is handled remotely from the Red Hat Enterprise Virtualization Manager. Remote authentication is used for user or API connections coming to the Manager from within an AD, IdM, or RHDS domain. The Red Hat Enterprise Virtualization Manager must be configured by an administrator using the **`engine-manage-domains`** tool to be a part of an RHDS, AD, or IdM domain. This requires that the Manager be provided with credentials for an account from the RHDS, AD, or IdM directory server for the domain with sufficient privileges to join a system to the domain. After domains have been added, domain users can be authenticated by the Red Hat Enterprise Virtualization Manager against the directory server using a password. The Manager uses a framework called the *Simple Authentication and Security Layer* (SASL) which in turn uses the *Generic Security Services Application Program Interface* (GSSAPI) to securely verify the identity of a user, and ascertain the authorization level available to the user.

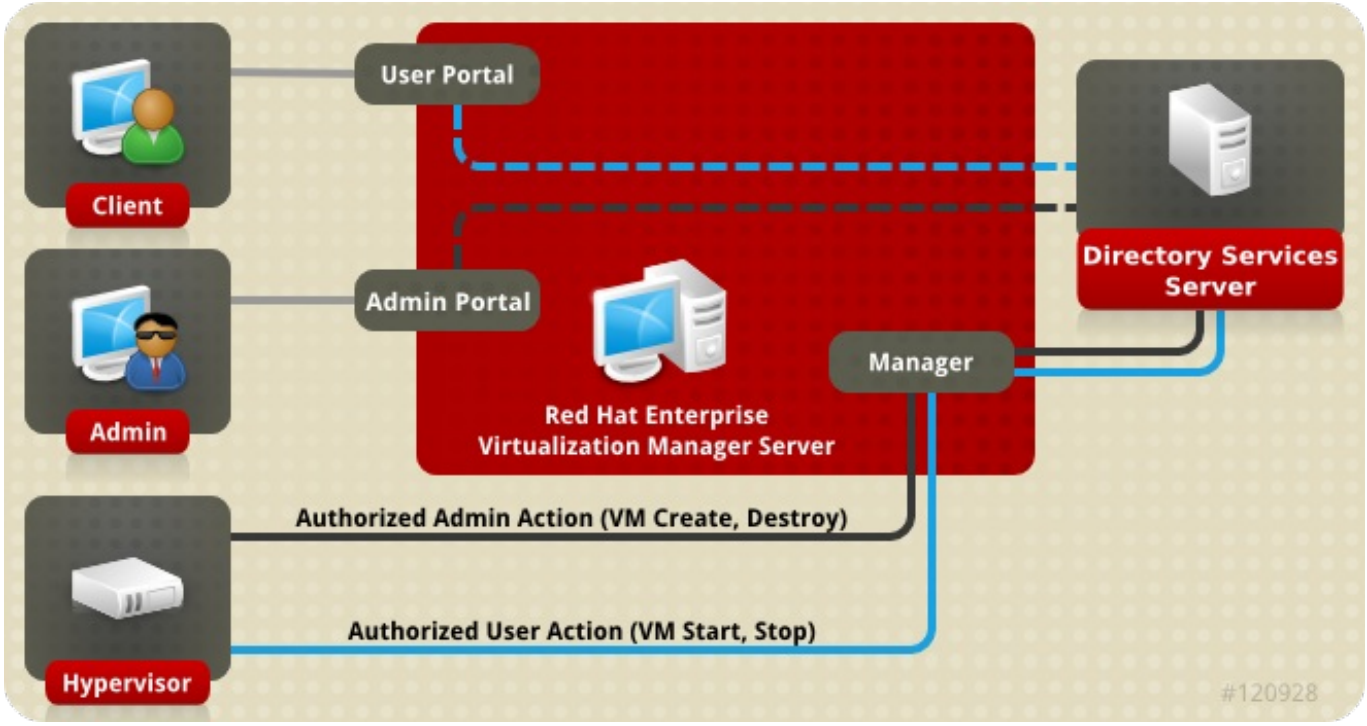


Figure 7.1. GSSAPI Authentication

Report a bug

Chapter 8. Templates and pools

8.1. Templates and Pools

The Red Hat Enterprise Virtualization environment provides administrators with tools to simplify the provisioning of virtual machines to users. These are *templates* and *pools*. A template is a shortcut that allows an administrator to quickly create a new virtual machine based on an existing, pre-configured virtual machine, bypassing operating system installation and configuration. This is especially helpful for virtual machines that will be used like appliances, for example web server virtual machines. If an organization uses many instances of a particular web server, an administrator can create a virtual machine that will be used as a template, installing an operating system, the web server, any supporting packages, and applying unique configuration changes. The administrator can then create a template based on the working virtual machine that will be used to create new, identical virtual machines as they are required.

Virtual machine pools are groups of virtual machines based on a given template that can be rapidly provisioned to users. Permission to use virtual machines in a pool is granted at the pool level; a user who is granted permission to use the pool will be assigned any virtual machine from the pool. Inherent in a virtual machine pool is the transitory nature of the virtual machines within it. Because users are assigned virtual machines without regard for which virtual machine in the pool they have used in the past, pools are not suited for purposes which require data persistence. Virtual machine pools are best suited for scenarios where either user data is stored in a central location and the virtual machine is a means to accessing and using that data, or data persistence is not important. The creation of a pool results in the creation of the virtual machines that populate the pool, in a stopped state. These are then started on user request.

Report a bug

8.2. Templates

To create a template, an administrator creates and customizes a virtual machine. Desired packages are installed, customized configurations are applied, the virtual machine is prepared for its intended purpose in order to minimize the changes that must be made to it after deployment. An optional but recommended step before creating a template from a virtual machine is *generalization*. Generalization is used to remove details like system user names, passwords, and timezone information that will change upon deployment. Generalization does not affect customized configurations. Generalization of Windows and Linux guests in the Red Hat Enterprise Virtualization environment is discussed it the *Red Hat Enterprise Virtualization Administration Guide*. Red Hat Enterprise Linux guests are generalized using **sys-unconfig**. Windows guests are generalized using **sys-prep**.

When the virtual machine that provides the basis for a template is satisfactorily configured, generalized if desired, and stopped, an administrator can create a template from the virtual machine. Creating a template from a virtual machine causes a read only copy of the specially configured virtual machine disk image to be created. The read only image will form the backing image for all subsequently created virtual machines that are based on that template. In other words, a template is essentially a customized read only disk image with an associated virtual hardware configuration. The hardware can be changed in virtual machines created from a template, for instance provisioning two gigabytes of RAM for a virtual machine created from a template that has one gigabyte of RAM. The template disk image, however, can not be changed as doing so would result in changes for all virtual machines based on the template.

When a template has been created, it can be used as the basis for multiple virtual machines. Virtual machines are created from a given template using a *Thin* provisioning method or a *Clone* provisioning method. Virtual machines that are cloned from templates take a complete writable copy of the template base image, sacrificing the space savings of a the thin creation method in exchange

for no longer depending on the presence of the template. Virtual machines that are created from a template using the thin method use the read only image from the template as a base image, requiring that the template and all virtual machines created from it be stored on the same storage domain. Changes to data and newly generated data are stored in a copy on write image. Each virtual machine based on a template uses the same base read only image, as well as a copy on write image that is unique to the virtual machine. This provides storage savings by limiting the number of times identical data is kept in storage. Furthermore, frequent use of the read only backing image can cause the data being accessed to be cached, resulting in a net performance increase.

Report a bug

8.3. Pools

Virtual machine pools allow for rapid provisioning of numerous identical virtual machines to users as desktops. Users who have been granted permission to access and use virtual machines from a pool receive an available virtual machine based on their position in a queue of requests. Virtual machines in a pool do not allow data persistence; each time a virtual machine is assigned from a pool, it is allocated in its base state. This is ideally suited to be used in situations where user data is stored centrally.

Virtual machine pools are created from a template. Each virtual machine in a pool uses the same backing read only image, and uses a temporary copy on write image to hold changed and newly generated data. Virtual machines in a pool are different from other virtual machines in that the copy on write layer that holds user generated and changed data is lost at shutdown. The implication of this is that a virtual machine pool requires no more storage than the template that backs it, plus some space for data generated or changed during use. Virtual machine pools are an efficient way to provide computing power to users for some tasks without the storage cost of providing each user with a dedicated virtual desktop.

> **Example 8.1. Example Pool Usage**
>
> A technical support company employs 10 help desk staff. However, only five are working at any given time. Instead of creating ten virtual machines, one for each help desk employee, a pool of five virtual machines can be created. Help desk employees allocate themselves a virtual machine at the beginning of their shift and return it to the pool at the end.

Report a bug

Chapter 9. Virtual Machine Snapshots

Snapshots are a storage function that allows an administrator to create a restore point of a virtual machine's operating system, applications, and data at a certain point in time. Snapshots save the data currently present in a virtual machine hard disk image as a COW volume and allow for a recovery to the data as it existed at the time the snapshot was taken. A snapshot causes a new COW layer to be created over the current layer. All write actions performed after a snapshot is taken are written to the new COW layer.

It is important to understand that a virtual machine hard disk image is a chain of one or more volumes. From the perspective of a virtual machine, these volumes appear as a single disk image. A virtual machine is oblivious to the fact that its disk is comprised of multiple volumes.

The term COW volume and COW layer are used interchangeably, however, layer more clearly recognizes the temporal nature of snapshots. Each snapshot is created to allow an administrator to discard unsatisfactory changes made to data *after* the snapshot is taken. Snapshots provide similar functionality to the **Undo** function present in many word processors.

Note

Snapshots of virtual machine hard disks marked **shareable** and those that are based on **Direct LUN** connections are not supported, live or otherwise.

The three primary snapshot operations are:

- Creation, which involves the first snapshot created for a virtual machine.
- Previews, which involves previewing a snapshot to determine whether or not to restore the system data to the point in time that the snapshot was taken.
- Deletion, which involves deleting a restoration point that is no longer required.

For task based information about snapshot operations, refer to the *Red Hat Enterprise Virtualization Administration Guide*.

Report a bug

9.1. Live Snapshots in Red Hat Enterprise Virtualization

In version 3.1, Red Hat Enterprise Virtualization introduced support for snapshots of running virtual machines.

Snapshots of virtual machine hard disks marked **shareable** and those that are based on **Direct LUN** connections are not supported, live or otherwise.

Any other virtual machine that is not being cloned or migrated can have a snapshot taken when running, paused, or stopped.

When a live snapshot of a virtual machine is initiated, the Manager requests that the SPM host create a new volume for the virtual machine to use. When the new volume is ready, the Manager uses VDSM to communicate with libvirt and qemu on the host running the virtual machine that it should begin using the new volume for virtual machine write operations. If the virtual machine is able to write to the new volume, the snapshot operation is considered a success and the virtual machine stops writing to the previous volume. If the virtual machine is unable to write to the new volume, the snapshot operation is considered a failure, and the new volume is deleted.

The virtual machine requires access to both its current volume and the new one from the time when a live snapshot is initiated until after the new volume is ready, so both volumes are opened with read-write access.

Virtual machines with a guest agent that supports quiescing installed can ensure filesystem consistency across snapshots. RHN registered Red Hat Enterprise Linux guests can install the **`qemu-guest-agent`** to enable quiescing before snapshots.

If a quiescing compatible guest agent is present on a virtual machine when it a snapshot is taken, VDSM uses libvirt to communicate with the agent to prepare for a snapshot. Outstanding write actions are completed, and then filesystems are frozen before a snapshot is taken. When the snapshot is complete, and libvirt has switched the virtual machine to the new volume for disk write actions, the filesystem is thawed, and writes to disk resume.

All live snapshots attempted with quiescing enabled. If the snapshot command fails because there is no compatible guest agent present, the live snapshot is re-initiated without the use-quiescing flag. When a virtual machine is reverted to its pre-snapshot state with quiesced filesystems, it boots cleanly with no filesystem check required. Reverting the previous snapshot using an un-quiesced filesystem requires a filesystem check on boot.

Report a bug

9.2. Snapshot Creation

In Red Hat Enterprise Virtualization the initial snapshot for a virtual machine is different from subsequent snapshots in that the initial snapshot retains its format, either QCOW2 or RAW. The first snapshot for a virtual machine designates existing volumes as a base image. Additional snapshots are additional COW layers tracking the changes made to the data stored in the image since the previous snapshot.

In Red Hat Enterprise Virtualization, a guest virtual machine usually interacts with a RAW disk image unless the image is created as a thinly provisioned image or the user specifically asked for it to be QCOW2. As depicted in Figure 9.1, “Initial Snapshot Creation”, the creation of a snapshot causes the volumes that comprise a virtual machine disk image to serve as the base image for all subsequent snapshots.

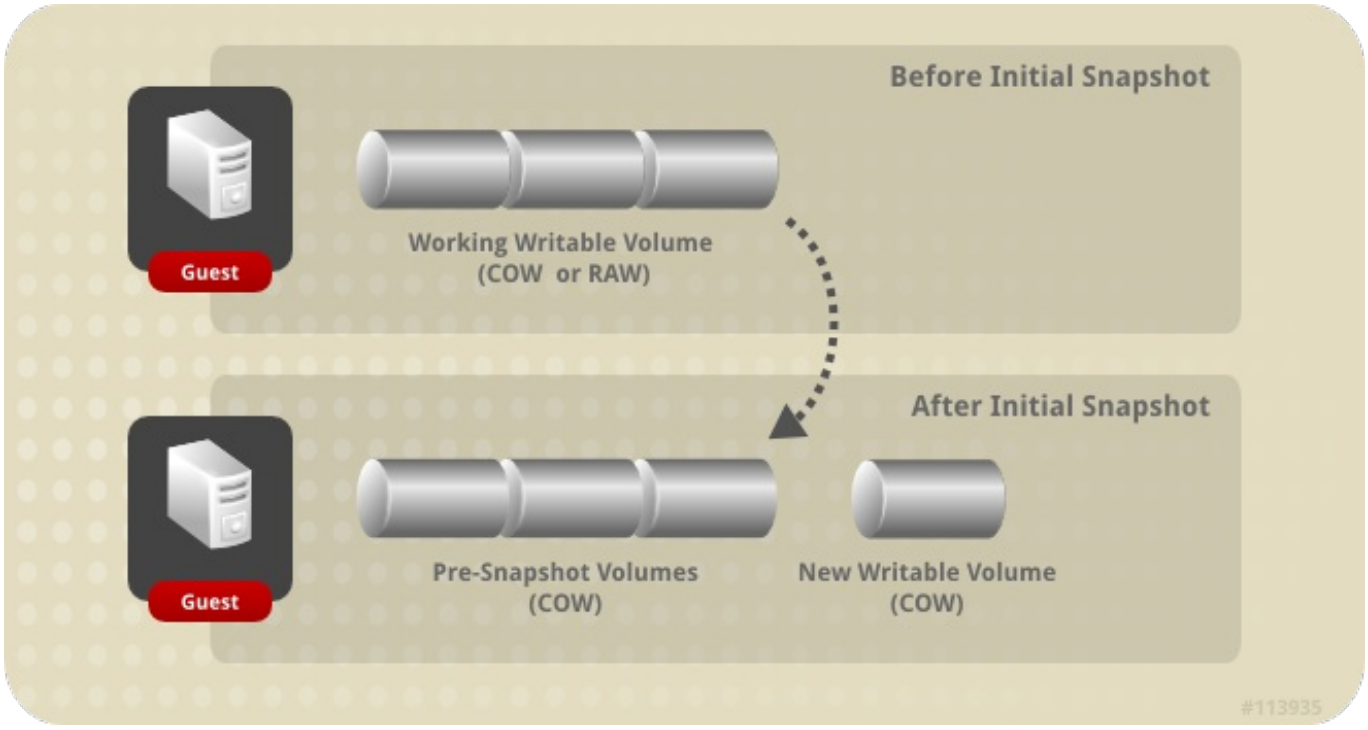


Figure 9.1. Initial Snapshot Creation

Snapshots taken after the initial snapshot result in the creation of new COW volumes in which data that is created or changed after the snapshot is taken will be stored. Each new COW layer begins containing only COW metadata. Data that is created through virtual machine use and operation after a snapshot is written to a new COW layer. When a virtual machine is used to modify data that exists in a previous COW layer, the data is read from the previous layer, and written into the newest layer. Virtual machines locate data by checking each COW layer from most recent to oldest, transparently to the virtual machine.

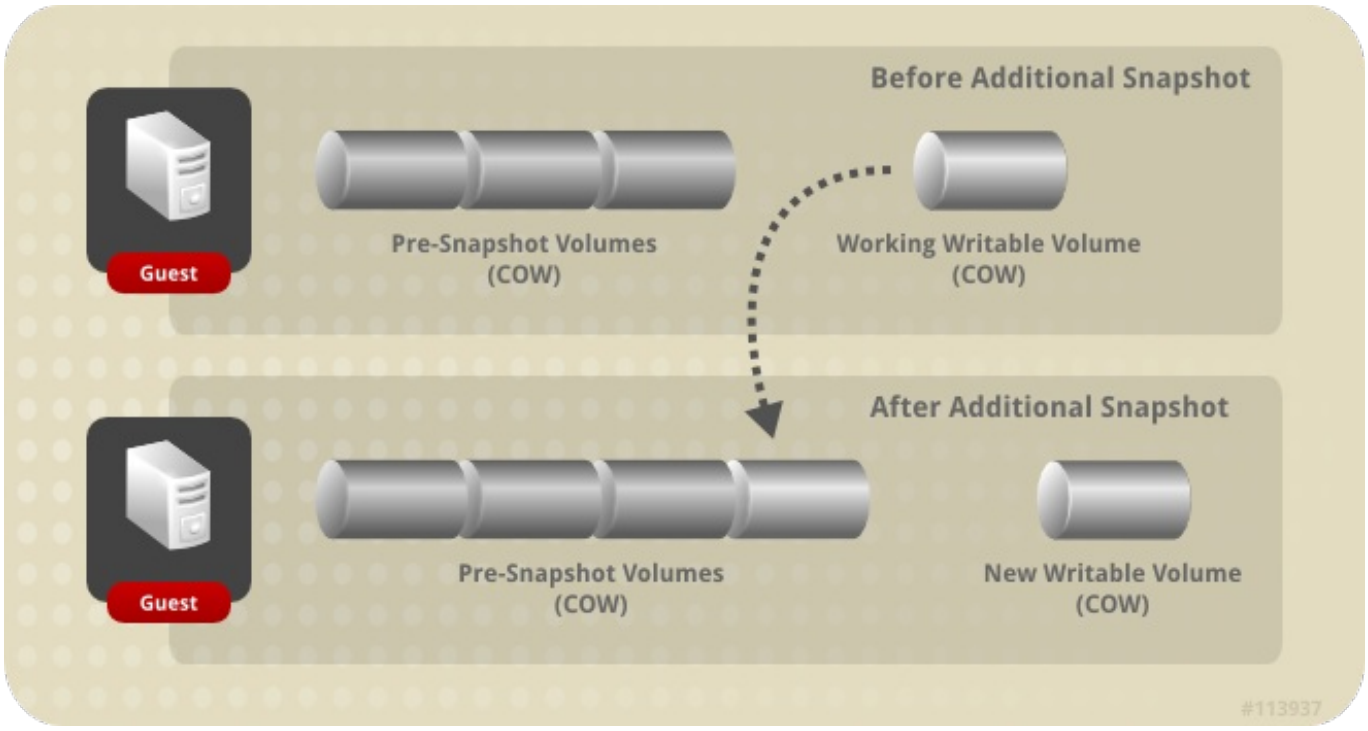


Figure 9.2. Additional Snapshot Creation

Report a bug

9.3. Snapshot Previews

To select which snapshot a virtual machine disk image will be reverted to, the administrator can preview all previously created snapshots.

From the available snapshots per guest, the administrator can select a snapshot volume to preview its contents. As depicted in Figure 9.3, “Preview Snapshot”, each snapshot is saved as a COW volume, and when it is previewed, a new preview layer is copied from the snapshot being previewed. The guest interacts with the preview instead of the actual snapshot volume.

After the administrator previews the selected snapshot, the preview can be committed to restore the guest data to the state captured in the snapshot. If the administrator commits the preview, the guest is attached to the preview layer.

After a snapshot is previewed, the administrator can select **Undo** to discard the preview layer of the viewed snapshot. The layer that contains the snapshot itself is preserved despite the preview layer being discarded.

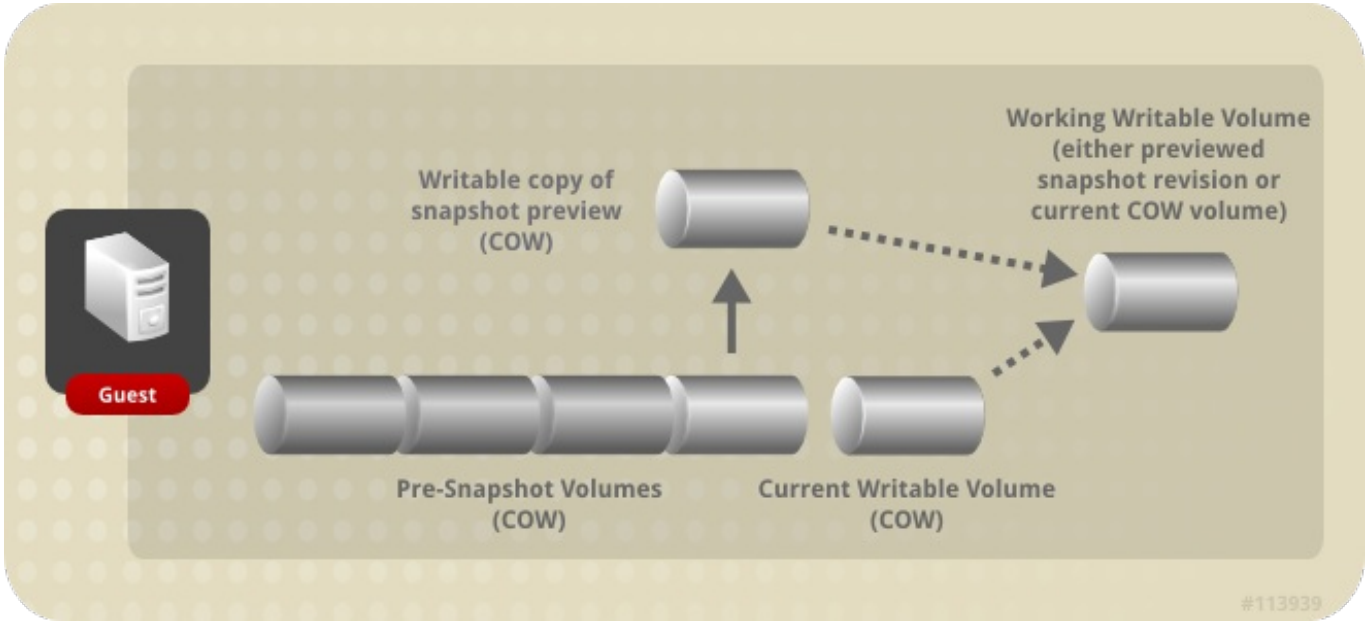


Figure 9.3. Preview Snapshot

Report a bug

9.4. Snapshot Deletion

If a snapshot or series of snapshots are no longer required, the administrator can delete one or more snapshots. The deletion of a snapshot does not necessarily cause the data in the snapshot to be deleted. For example, if the third snapshot out of five snapshots is deleted, the unchanged data in the third snapshot must be preserved for the fourth and fifth snapshots to be usable. Snapshot deletion is not an operation to preserve storage capacity within the Red Hat Enterprise Virtualization environment. Snapshot deletion allows an administrator to remove a potential data restoration point when it becomes clear that it will not be necessary to return a virtual machine hard disk image data to the point in time the snapshot preserves.

When the administrator deletes a snapshot, the data from the deleted snapshot and the snapshot created after the deleted snapshot are merged into a single COW volume. After the two snapshots are merged, the resultant volume contains any data that was created or modified prior to the deleted snapshot and after the deleted snapshot. No data has been removed, only the ability to restore a point in time in the life of the virtual machine hard disk image. As displayed in Figure 9.4, “Snapshot Deletion”, snapshot 2 is selected for deletion. As a consequence, snapshot 2 and snapshot 3 are merged, saving the changes in both snapshots in the COW volume for snapshot 3 (i.e. the newer snapshot) as the replacement for the deleted snapshot.

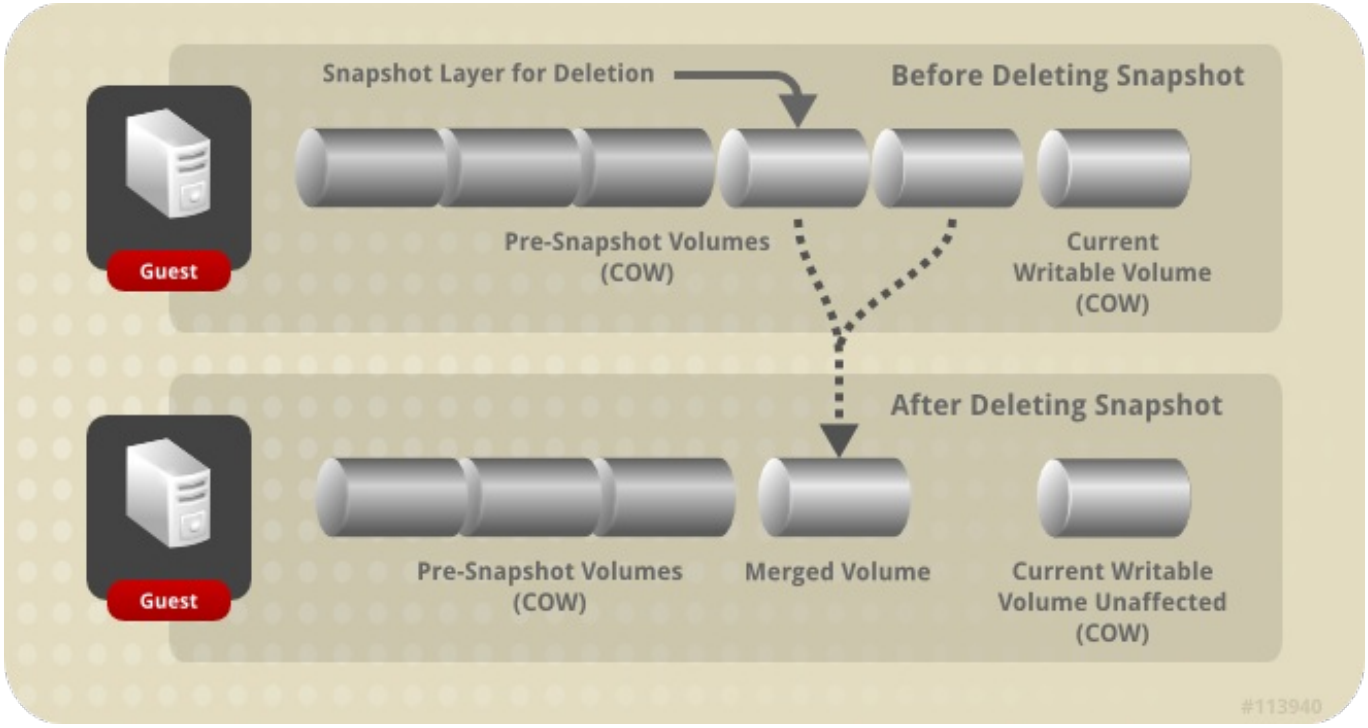


Figure 9.4. Snapshot Deletion

Report a bug

Chapter 10. Hardware drivers and devices

10.1. Virtualized Hardware

Red Hat Enterprise Virtualization presents three distinct types of system devices to virtualized guests. These hardware devices all appear as physically attached hardware devices to the virtualized guest but the device drivers work in different ways.

Emulated devices

Emulated devices, sometimes referred to as *virtual devices*, exist entirely in software. *Emulated device drivers* are a translation layer between the operating system running on the host (which manages the source device) and the operating systems running on the guests. The device level instructions directed to and from the emulated device are intercepted and translated by the hypervisor. Any device of the same type as that being emulated and recognized by the Linux kernel is able to be used as the backing source device for the emulated drivers.

Para-virtualized Devices

Para-virtualized devices require the installation of device drivers on the guest operating system providing it with an interface to communicate with the hypervisor on the host machine. This interface is used to allow traditionally intensive tasks such as disk I/O to be performed outside of the virtualized environment. Lowering the overhead inherent in virtualization in this manner is intended to allow guest operating system performance closer to that expected when running directly on physical hardware.

Physically shared devices

Certain hardware platforms allow virtualized guests to directly access various hardware devices and components. This process in virtualization is known as *passthrough* or *device assignment*. Passthrough allows devices to appear and behave as if they were physically attached to the guest operating system.

Report a bug

10.2. Stable Device Addresses in Red Hat Enterprise Virtualization

Prior to Red Hat Enterprise Virtualization 3.1, the PCI addresses of virtual machine hardware devices were allocated in the order in which the devices were discovered. This meant that if the order in which virtual hardware was discovered changed, the PCI address allocation given to the hardware could also change.

A change in PCI device addresses is particularly detrimental to virtual machines running Windows operating systems. If an important device, like a system hard disk, were to be allocated a different PCI address than the one that Windows was expecting, Windows anti-piracy measures could require a re-activation of the operating system.

Beginning in Red Hat Enterprise Virtualization 3.1, virtual hardware PCI address allocations are persisted in the ovirt-engine database.

PCI addresses are allocated by **QEMU** at virtual machine creation time, and reported to **VDSM** by **libvirt**. **VDSM** reports them back to the Manager, where they are stored in the ovirt-engine database.

When a virtual machine is started, the Manager sends **VDSM** the device address out of the database. **VDSM** passes them to **libvirt** which starts the virtual machine using the PCI device addresses that were allocated when the virtual machine was run for the first time.

When a device is removed from a virtual machine, all references to it, including the stable PCI address, are also removed. If a device is added to replace the removed device, it is allocated a PCI address by **QEMU**, which is unlikely to be the same as the device it replaced.

Report a bug

10.3. Central Processing Unit (CPU)

Each Red Hat Enterprise Virtualization Hypervisor within a Cluster has a number of *virtual CPUs* (vCPUS). The virtual CPUs are in turn exposed to guests running on the hypervisors. All virtual CPUs exposed by Hypervisors within a Cluster are of the type selected when the Cluster was initially created via Red Hat Enterprise Virtualization Manager. Mixing of virtual CPU types within a Cluster is not possible.

Each available virtual CPU type has characteristics based on physical CPUs of the same name. The virtual CPU is indistinguishable from the physical CPU to the guest operating system.

Note

Support for *x2APIC:*

All virtual CPU models provided by Red Hat Enterprise Linux 6 hosts include support for x2APIC. This provides an *Advanced Programmable Interrupt Controller (APIC)* to better handle hardware interrupts.

Report a bug

10.4. System Devices

System devices are critical for the guest to run and cannot be removed. Each system device attached to a guest also takes up an available PCI slot. The default system devices are:

- the host bridge,
- the ISA bridge and USB bridge (The USB and ISA bridges are the same device),
- the graphics card (using either the Cirrus or qxl driver), and
- the memory balloon device.

Report a bug

10.5. Network Devices

Red Hat Enterprise Virtualization is able to expose three different types of network interface controller to guests. The type of network interface controller to expose to a guest is chosen when the guest is created but is changeable from the Red Hat Enterprise Virtualization Manager.

- The **e1000** network interface controller exposes a virtualized Intel PRO/1000 (e1000) to guests.
- The **virtio** network interface controller exposes a para-virtualized network device to guests.
- The **rtl8139** network interface controller exposes a virtualized **Realtek Semiconductor Corp RTL8139** to guests.

Multiple network interface controllers are permitted per guest. Each controller added takes up an available PCI slot on the guest.

Report a bug

10.6. Graphics Devices

Two emulated graphics devices are provided. These devices can be connected to with the SPICE protocol or with VNC.

- The **ac97** emulates a **Cirrus CLGD 5446 PCI VGA** card.
- The **vga** emulates a dummy VGA card with **Bochs**VESA extensions (hardware level, including all non-standard modes).

Report a bug

10.7. Storage Devices

Storage devices and storage pools can use the block device drivers to attach storage devices to virtualized guests. Note that the storage drivers are not storage devices. The drivers are used to attach a backing storage device, file or storage pool volume to a virtualized guest. The backing storage device can be any supported type of storage device, file, or storage pool volume.

- The **IDE** driver exposes an emulated block device to guests. The emulated **IDE** driver can be used to attach any combination of up to four virtualized **IDE** hard disks or virtualized **IDE** CD-ROM drives to each virtualized guest. The emulated **IDE** driver is also used to provide virtualized DVD-ROM drives.
- The **VirtIO** driver exposes a para-virtualized block device to guests. The para-virtualized block driver is a driver for all storage devices supported by the hypervisor attached to the virtualized guest (except for floppy disk drives, which must be emulated).

Report a bug

10.8. Sound Devices

Two emulated sound devices are available:

- The **ac97** emulates an **Intel 82801AA AC97 Audio** compatible sound card.
- The **es1370** emulates an **ENSONIQ AudioPCI ES1370** sound card.

Report a bug

10.9. Serial Driver

The para-virtualized serial driver (**`virtio-serial`**) is a bytestream-oriented, character stream driver. The para-virtualized serial driver provides a simple communication interface between the host's user space and the guest's user space where networking is not be available or unusable.

Report a bug

10.10. Balloon Driver

The balloon driver allows guests to express to the hypervisor how much memory they require. The balloon driver allows the host to efficiently allocate and memory to the guest and allow free memory to be allocated to other guests and processes.

Guests using the balloon driver can mark sections of the guest's RAM as not in use (balloon inflation). The hypervisor can free the memory and use the memory for other host processes or other guests on that host. When the guest requires the freed memory again, the hypervisor can reallocate RAM to the guest (balloon deflation).

Report a bug

Chapter 11. Technical minimums and limitations

11.1. Minimum Requirements and Supported Limits

There are a number of physical and logical limitations which apply to Red Hat Enterprise Virtualization environments. Environments with configurations outside of these limitations are currently not supported.

Report a bug

11.2. Data Center Limitations

In a managed virtual environment the highest level container for all resources is the data center. A number of limitations apply to the resources which can be contained within each data center.

Table 11.1. data center Limitations

Item	Limitations
Number of storage domains	A minimum of 2 storage domains per data center is recommended. One data storage domain is required, and an ISO storage domain per data center is recommended.
Number of hosts	A maximum of 200 hosts per data center is supported.

Report a bug

11.3. Cluster Limitations

A cluster is a set of physical hosts that are treated as a resource pool for a set of virtual machines. Hosts in a cluster share the same network infrastructure and the same storage. The cluster is a migration domain within which virtual machines can be moved from host to host. To ensure stability a number of limitations apply to each cluster.

- All managed hypervisors must be in a cluster.
- All managed hypervisors within a cluster must have the same CPU type. Intel and AMD CPUs cannot co-exist within the same cluster.

Note

Further information about clusters is available in the *Red Hat Enterprise Virtualization Administration Guide*.

Report a bug

11.4. Storage Domain Limitations

Storage domains provide space for the storage of virtual machine disk images and ISO images as well as the import and export of virtual machines. While many storage domains may be created within a given data center there are a number of limitations and recommendations that apply to each storage domain.

Table 11.2. storage domain Limitations

Item	Limitations

Storage Types	- Supported storage types are:
	  - Fibre Channel Protocol (FCP)
	  - Internet Small Computer System Interface (iSCSI)
	  - Network File System (NFS)
	- All data storage domains within a data center must be of the same type. The type is specified as a step in the creation of the storage domain.
	  - The data storage domain can be any of FCP, iSCSI, and NFS
	  - Legacy FCP or iSCSI export storage domains from Red Hat Enterprise Virtualization 2.2 environments can be attached to data centers in Red Hat Enterprise Virtualization 3.0. New ISO and export storage domains must be provided by NFS.

Logical Unit Numbers (LUNs)	- No more than 300 LUNs are permitted for each storage domain that is provided by iSCSI or FCP.

Item	Limitations
Logical Volumes (LVs)	In Red Hat Enterprise Virtualization, logical volumes represent virtual disks for virtual machines, templates, and virtual machine snapshots. • No more than 350 logical volumes are recommended for each storage domain that is provided by iSCSI or FCP. If the number of logical volumes in a given storage domain exceeds this number, splitting available storage into separate storage domains with no more than 350 logical volumes each is recommended. The root cause of this limitation is the size of LVM metadata. As the number of logical volumes increases, the LVM metadata associated with those logical volumes also increases. When this metadata exceeds 1 MB in size, the performance of provisioning operations such as creating new disks or snapshots decreases, and lvextend operations for thinly provisioning a logical volume when running a qcow disk take a longer time to run.

Note

Further information about storage domains is available in the *Red Hat Enterprise Virtualization Administration Guide*.

Report a bug

11.5. Red Hat Enterprise Virtualization Manager Limitations

Red Hat Enterprise Virtualization Manager servers must run Red Hat Enterprise Linux 6. A number of additional hardware requirements must also be met.

Table 11.3. Red Hat Enterprise Virtualization Manager Limitations

Item	Limitations
RAM	• A minimum of 3 GB of RAM is required.
PCI Devices	• At least one network controller with a minimum bandwidth of 1 Gbps is recommended.

Item	Limitations
Storage	A minimum of 3 GB of available local disk space is recommended.

Note

Further information about Red Hat Enterprise Virtualization Manager is available in the *Red Hat Enterprise Virtualization Installation Guide*.

Report a bug

11.6. Hypervisor Requirements

Red Hat Enterprise Virtualization Hypervisors have a number of hardware requirements and supported limits.

Table 11.4. Red Hat Enterprise Virtualization Hypervisor Requirements and Supported Limits

Item	Support Limit
CPU	A minimum of 1 physical CPU is required. Red Hat Enterprise Virtualization supports the use of these CPU models in virtualization hosts: - AMD Opteron G1 - AMD Opteron G2 - AMD Opteron G3 - AMD Opteron G4 - AMD Opteron G5 - Intel Conroe - Intel Penryn - Intel Nehalem - Intel Westmere - Intel Haswell All CPUs must have support for the Intel® 64 or AMD64 CPU extensions, and the AMD-V™ or Intel VT® hardware virtualization extensions enabled. Support for the **No eXecute** flag (NX) is also required.

Item	Support Limit

RAM	• A minimum of 2 GB of RAM is recommended.
	• The amount of RAM required for each virtual machine varies depending on:
	  ◦ guest operating system requirements,
	  ◦ guest application requirements, and
	  ◦ memory activity and usage of virtual machines.
	
	Additionally KVM is able to over-commit physical RAM for virtual machines. It does this by only allocating RAM for virtual machines as required and shifting underutilized virtual machines into swap.
	
	• A maximum of 2 TB of RAM is supported.

Item	Support Limit

Storage	The minimum supported internal storage for a Hypervisor is the total of the following list:
	- The root partitions require at least 512 MB of storage.
	- The configuration partition requires at least 8 MB of storage.
	- The recommended minimum size of the logging partition is 2048 MB.
	- The data partition requires at least 256 MB of storage. Use of a smaller data partition may prevent future upgrades of the Hypervisor from the Red Hat Enterprise Virtualization Manager. By default all disk space remaining after allocation of swap space will be allocated to the data partition.
	- The swap partition requires at least 8 MB of storage. The recommended size of the swap partition varies depending on both the system the Hypervisor is being installed upon and the anticipated level of overcommit for the environment. Overcommit allows the Red Hat Enterprise Virtualization environment to present more RAM to virtual machines than is actually physically present. The default overcommit ratio is **0.5**.
	
	The recommended size of the swap partition can be determined by:
	  - Multiplying the amount of system RAM by the expected overcommit ratio, and adding
	  - 2 GB of swap space for systems with 4 GB of RAM or less, or
	  - 4 GB of swap space for systems with between 4 GB and 16 GB of RAM, or
	  - 8 GB of swap space for systems with between 16 GB and 64 GB of RAM, or
	  - 16 GB of swap space for systems with between 64 GB and 256 GB of RAM.
	
	Example 11.1. Calculating Swap Partition Size
	
	For a system with 8 GB of RAM this means the formula for determining the amount of swap space to allocate is:
	
	`(8 GB x 0.5) + 4 GB = 8 GB`

Item	Support Limit
PCI Devices	Please note that these are the *minimum* storage requirements for Hypervisor installation. It is recommended to use the default allocations which use more storage space. • At least one network controller is required with a recommended minimum bandwidth of 1 Gbps.

Important

When the Red Hat Enterprise Virtualization Hypervisor boots a message may appear:

```
Virtualization hardware is unavailable.
(No virtualization hardware was detected on this system)
```

This warning indicates the virtualization extensions are either disabled or not present on your processor. Ensure that the CPU supports the listed extensions and they are enabled in the system BIOS.

To check that processor has virtualization extensions, and that they are enabled:

- At the Hypervisor boot screen press any key and select the **Boot** or **Boot with serial console** entry from the list. Press **Tab** to edit the kernel parameters for the selected option. After the last kernel parameter listed ensure there is a **Space** and append the **rescue** parameter.
- Press **Enter** to boot into rescue mode.
- At the prompt which appears, determine that your processor has the virtualization extensions and that they are enabled by running this command:

  ```
  # grep -E 'svm|vmx' /proc/cpuinfo
  ```

  If any output is shown, the processor is hardware virtualization capable. If no output is shown it is still possible that your processor supports hardware virtualization. In some circumstances manufacturers disable the virtualization extensions in the BIOS. Where you believe this to be the case consult the system's BIOS and the motherboard manual provided by the manufacturer.

- As an additional check, verify that the **kvm** modules are loaded in the kernel:

  ```
  # lsmod | grep kvm
  ```

  If the output includes **kvm_intel** or **kvm_amd** then the **kvm** hardware virtualization modules are loaded and your system meets requirements.

Important

The Red Hat Enterprise Virtualization Hypervisor does not support installation on `fakeraid` devices. Where a `fakeraid` device is present it must be reconfigured such that it no longer runs in RAID mode.

1. Access the RAID controller's BIOS and remove all logical drives from it.
2. Change controller mode to be non-RAID. This may be referred to as compatibility or JBOD mode.

Access the manufacturer provided documentation for further information related to the specific device in use.

Report a bug

11.7. Guest Requirements and Support Limits

The following requirements and support limits apply to guests that are run on the Hypervisor:

Table 11.5. Virtualized Hardware

Item	Limitations
CPU	A maximum of 160 virtualized CPUs per guest is supported.

Item	Limitations
RAM	Different guests have different RAM requirements. The amount of RAM required for each guest varies based on the requirements of the guest operating system and the load under which the guest is operating. A number of support limits also apply. - A minimum of 512 MB of virtualized RAM per guest is supported. Creation of guests with less than 512 MB of virtualized RAM while possible is not supported. - A maximum of 512 GB of virtualized RAM per 64 bit guest is supported. - The supported virtualized RAM maximum for 32 bit virtual machines varies depending on the virtual machine. 32 bit virtual machines operating in standard 32 bit mode have a supported virtualized RAM maximum of 4 GB virtualized RAM per virtual machine. However, note that some virtualized operating systems will only use 2 GB of the supported 4 GB. 32 bit virtual machines operating in PAE (Page Address Extension) mode have a supported virtualized RAM maximum of 64 GB per virtual machine. However, not all virtualized operating systems can be configured to use this amount of virtualized RAM.
PCI devices	- A maximum of 31 virtualized PCI devices per guest is supported. A number of system devices count against this limit, some of which are mandatory. Mandatory devices which count against the PCI devices limit include the PCI host bridge, ISA bridge, USB bridge, board bridge, graphics card, and the IDE or VirtIO block device.
Storage	- A maximum of 28 virtualized storage devices per guest is supported, composed of a possible 3 IDE and 25 Virtio.

Report a bug

11.8. SPICE Limitations

SPICE currently supports a maximum resolution of 2560x1600 pixels.

Report a bug

Chapter 12. Additional References

12.1. Additional References

These additional documentation resources do not form part of the Red Hat Enterprise Virtualization documentation suite. They do however contain useful information for System Administrators managing Red Hat Enterprise Virtualization environments and are available at https://access.redhat.com/documentation/en-US.

Red Hat Enterprise Linux - Deployment Guide

A guide to the deployment, configuration and administration of Red Hat Enterprise Linux.

Red Hat Enterprise Linux - DM-Multipath Guide

A guide to the use of Device-Mapper Multipathing on Red Hat Enterprise Linux.

Red Hat Enterprise Linux - Installation Guide

A guide to the installation of Red Hat Enterprise Linux.

Red Hat Enterprise Linux - Storage Administration Guide

A guide to the management of storage devices and file systems on Red Hat Enterprise Linux.

Red Hat Enterprise Linux - Virtualization Administration Guide

A guide to the installation, configuration, administration and troubleshooting of virtualization technologies in Red Hat Enterprise Linux.

Report a bug

Revision History

Revision 3.3-13 **Fri 20 Mar 2015** **Tahlia Richardson**

BZ#1203488 - Removed references to now-defunct Hypervisor Deployment Guide.

Revision 3.3-12.1 **Tue 29 Apr 2014** **Andrew Burden**

Publishing for 3.3.z release.

Revision 3.3-12 **Wed 29 Jan 2014** **Andrew Dahms**

BZ#1059072 - Changed the prefix for command line utilities from 'rhevm-' to 'engine-'.

Revision 3.3-11 **Wed 18 Dec 2013** **Andrew Dahms**

BZ#1039186 - Added an explanation of the limitation in number of logical volumes per volume group.

Revision 3.3-10 **Wed 18 Dec 2013** **Andrew Dahms**

BZ#1023541 - Removed a line describing the deletion of data when the latest snapshot in a series is deleted.

Revision 3.3-9 **Tue 15 Oct 2013** **Zac Dover**

BZ#978154 - Support ISO domain on any file storage

Revision 3.3-8 **Tue 15 Oct 2013** **Red Hat Engineering Content Services**

Updating for public beta.

Revision 3.3-7 **Wed 09 Oct 2013** **Zac Dover**

BZ#978874 - pNFS Client Support - bumping the revnumber so that the book travels through the buildsystem

Revision 3.3-6 **Fri 04 Oct 2013** **Red Hat Engineering Content Services**

Fixed typos in the "Features Requiring a Compatibility Upgrade to Red Hat Enterprise Virtualization 3.3" topic.

Revision 3.3-5 **Mon 30 Sep 2013** **Red Hat Engineering Content Services**

BZ#976251 - Finished table of features requiring a compatibility upgrade to Red Hat Enterprise Virtualization 3.3
BZ#976251 - added additonal features to the list of features requiring compatibility upgrade.
BZ#976251 - added descriptions to the features requiring compatibility upgrade.
BZ#976251 - Added summary of features that require a compatibility upgrade.

Revision 3.3-4 **Fri 23 Aug 2013** **Zac Dover**

BZ#978874 - pNFS support now included out-of-the-box in RHEL 6.4
BZ#978185 - OpenLDAP now supported as a Directory Service
BZ#978737 - Soft-Fencing Hosts over SSH
BZ#978813 - Changing "JBoss EAP" to "Red Hat JBoss Enterprise Application Platform
BZ#979256 - Describing fencing behavior in required and non-required networks.
BZ#978154 - ISO domains are now supported on all file storage types.
BZ#978801 - Management-related network traffic is now segregated from migration-related network traffic.

Revision 3.3-3	Tue 20 Aug 2013	Andrew Burden

BZ#976608 - Topics regarding SPM updated to include manually setting the SPM.

Revision 3.3-2	Thu 25 Jul 2013	Red Hat Engineering Content Services

Removed sentence in "Power Management by Proxy" mentioning the introduction of a feature in 3.2 as it is no longer a new feature.

Revision 3.3-1	Thu 18 Jul 2013	Tim Hildred

Initial creation for 3.3 release.